Mathematics of Data **Fall 2019**

Homework #3 - Due date: 6th December 2019

Student: Oriol Barbany Mayor

Problem 1 - Multiclass Classification

Part I - Theory

1.I.1 Assuming that the samples are statistically independent, we can write the likelihood of observing vector $\mathbf{b} \in \{1, \dots, C\}^n$ as

$$\Pr(\mathbf{b}|A, X) = \prod_{i=1}^{n} \prod_{j=1}^{C} \Pr(b_i = j|\mathbf{a}_i, X)^{\mathbb{1}_{\{b_i=j\}}} = \prod_{i=1}^{n} \prod_{j=1}^{C} \left(\frac{e^{\mathbf{a}_i^T \mathbf{x}_j}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_j}} \right)^{\mathbb{1}_{\{b_i=j\}}} \tag{1}$$

The log-likelihood thus have the form

$$\ell(X) := \log \Pr(\mathbf{b}|A, X) = \sum_{i=1}^{n} \sum_{j=1}^{C} \mathbb{1}_{\{b_i=j\}} \left(\mathbf{a}_i^T \mathbf{x}_j - \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_j} \right) = \sum_{i=1}^{n} \left(\mathbf{a}_i^T \mathbf{x}_{b_i} - \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} \right) \tag{2}$$

and the maximum likelihood estimator of X is given by

$$\hat{X}_{ML} \in \arg\max_X \{\ell(X)\} = \arg\min_X \left\{ -\ell(X) := \sum_{i=1}^{n} \left(\log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} - \mathbf{a}_i^T \mathbf{x}_{b_i} \right) =: f(X) \right\} \tag{3}$$

1.I.2 Following the proposed notation, the $i-$th column of matrix X is denoted as $\mathbf{x}_i$. We can write the gradient of X as

$$\nabla_X f(X) = [\nabla_{\mathbf{x}_1} f(X), \cdots, \nabla_{\mathbf{x}_C} f(X)] \tag{4}$$

Let's find the value of this gradient at one of the columns, namely j

$$\nabla_{\mathbf{x}_j} f(X) = \sum_{i=1}^{n} \left(\frac{\mathbf{a}_i e^{\mathbf{a}_i^T \mathbf{x}_j}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} - \mathbf{a}_i \mathbb{1}_{\{b_i=j\}} \right) = \sum_{i=1}^{n} \mathbf{a}_i \left(\frac{e^{\mathbf{a}_i^T \mathbf{x}_j}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} - \mathbb{1}_{\{b_i=j\}} \right) \tag{5}$$

In matrix notation, we can write

$$\nabla_X f(X) = A^T (Z \exp(AX) - Y) \tag{6}$$

where $Z \in \mathbb{R}^{n \times n}$ has entries

$$Z_{i,j} = \begin{cases} \frac{1}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases},$$

$Y \in \{0, 1\}^{n \times C}$ is a matrix whose rows are the one-hot encoding of b_i and exp is applied entry-wise.

1.I.3 Proving that f is convex is equivalent to showing that its Hessian is PSD. The hessian of f has the form

$$\nabla^2 f(X) = \sum_{i=1}^{n} \Sigma_i \otimes \mathbf{a}_i \mathbf{a}_i^T \tag{7}$$

Fact 1. *If $D, E \succeq 0$, then $D \otimes E \succeq 0$.*

So to proof that $\nabla^2 f(X)$ is PSD, it's only left to show that $D, E \succeq 0 \implies D + E \succeq 0$ and $\mathbf{a}_i \mathbf{a}_i^T, \Sigma_i \succeq 0$. By definition,

$$D \succeq 0 \iff \mathbf{z}^T D \mathbf{z} \geq 0 \quad \forall \mathbf{z} \tag{8}$$

so if $D, E \succeq 0$,

$$\mathbf{z}^T (D + E)\mathbf{z} = \mathbf{z}^T D \mathbf{z} + \mathbf{z}^T E \mathbf{z} \geq \mathbf{z}^T E \mathbf{z} \geq 0 \quad \forall \mathbf{z} \iff D + E \succeq 0 \tag{9}$$

where the inequalities follow from positive semidefiniteness of D and E respectively.

Note that $\mathbf{a}_i \mathbf{a}_i^T$ is indeed PSD, since

$$\mathbf{z}^T \mathbf{a}_i \mathbf{a}_i^T \mathbf{z} = \left\| \mathbf{a}_i^T \mathbf{z} \right\|^2 \geq 0 \quad \forall \mathbf{z} \iff \mathbf{a}_i \mathbf{a}_i^T \succeq 0 \tag{10}$$

where the inequality follows by non-negativity of the norm.

Finally, to show that Σ_i is also PSD, note that $\Sigma_i = \Lambda - \mathbf{s}_i \mathbf{s}_i^T$ with $\mathbf{s}_i = [\sigma_{i1}, \ldots, \sigma_{iC}]^T$ and $\Lambda = \text{diag}(\mathbf{s}_i)$. By definition of positive semidefiniteness,

$$\begin{aligned}
\mathbf{z}^T \Sigma_i \mathbf{z} &= \mathbf{z}^T (\Lambda - \mathbf{s}_i \mathbf{s}_i^T)\mathbf{z} = \mathbf{z}^T \Lambda \mathbf{z} - \left\| \mathbf{s}_i^T \mathbf{z} \right\|^2 && (11)\\
&:= \sum_{j=1}^{C} z_j^2 \sigma_{ij} - \left(\sum_{j=1}^{C} z_j \sigma_{ij} \right)^2 && (12)\\
&\geq \sum_{j=1}^{C} z_j^2 \sigma_{ij} - \sum_{j=1}^{C} (z_j \sigma_{ij})^2 && \text{by convexity of } (\cdot)^2 \quad (13)\\
&= \sum_{j=1}^{C} z_j^2 (\sigma_{ij} - \sigma_{ij}^2) \geq \sum_{j=1}^{C} z_j^2 && \text{since } 0 \leq \sigma_{ij} \leq 1 \quad (14)\\
&:= \|\mathbf{z}\| \geq 0 && \text{by non-negativity of norm} \quad (15)
\end{aligned}$$

The proof concludes by applying Fact 1 to show that $\Sigma_i \otimes \mathbf{a}_i \mathbf{a}_i^T \succeq 0 \quad \forall i$ and by again applying (9) n times to show $\sum_i \Sigma_i + \mathbf{a}_i \mathbf{a}_i^T \succeq 0$.

1.I.4 We can upperbound the largest eigenvalue of the matrix $\mathbf{a}_i \mathbf{a}_i^T$ by expressing it as a maximization problem

$$\begin{aligned}
\lambda_{\max}(\mathbf{a}_i \mathbf{a}_i^T) &= \max_{\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\mathbf{x}^T \mathbf{a}_i \mathbf{a}_i^T \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \max_{\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\mathbf{x}^T \mathbf{a}_i \mathbf{a}_i^T \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \max_{\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\langle \mathbf{x}, \mathbf{a}_i \rangle^2}{\|\mathbf{x}\|^2} && (16)\\
&\leq \max_{\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\|\mathbf{x}\|^2 \|\mathbf{a}_i\|^2}{\|\mathbf{x}\|^2} = \|\mathbf{a}_i\|^2 && (17)
\end{aligned}$$

where inequality follows from Cauchy-Schwarz.

Fact 2. *Given $D \succeq 0$, we have that*

$$\lambda_{\max}(D) \leq \max_i \sum_j |D_{ij}| \tag{18}$$

that is, the top eigenvalue is upper-bounded by the maximum ℓ_1 norm of rows.

Using Fact 2 for the PSD matrix Σ_i, yields

$$\lambda_{\max}(\Sigma_i) \leq \max_j \sum_{k=1}^{C} |\Sigma_{i_{jk}}| = \max_j \sigma_{ij} \left[(1-\sigma_{ij}) + \sum_{k=1, k\neq j}^{C} \sigma_{ik} \right] \tag{19}$$

where I use $1 \geq \sigma_{ij} \geq 0 \quad \forall i, j \in \{1, \ldots, C\}$ to get rid of the absolute value.
Note that $1 - \sigma_{ij} = \sum_{k=1, k\neq j}^{C} \sigma_{ik}$ since $\sum_k \sigma_{ik} = 1 \ \forall i$, so

$$\lambda_{\max}(\Sigma_i) \leq \max_j 2\sigma_{ij}(1-\sigma_{ij}) \leq \frac{1}{2} \tag{20}$$

where the last inequality follows by maximizing the resulting expression, i.e.

$$\frac{\partial}{\partial \sigma_{ij}} 2\sigma_{ij}(1-\sigma_{ij}) = 2(1-2\sigma_{ij}) = 0 \iff \sigma_{ij} = \frac{1}{2} \tag{21}$$

Fact 3. *If $D, E \succeq 0$, then $\lambda_{\max}(D \otimes E) = \lambda_{\max}(D)\lambda_{\max}(E)$.*

Using Fact 3, we have that

$$\lambda_{\max}(\Sigma_i \otimes \mathbf{a}_i \mathbf{a}_i^T) = \lambda_{\max}(\Sigma_i)\lambda_{\max}(\mathbf{a}_i \mathbf{a}_i^T) \leq \frac{1}{2}\lambda_{\max}(\mathbf{a}_i \mathbf{a}_i^T) \leq \frac{\|\mathbf{a}_i\|^2}{2} \tag{22}$$

Lemma 1. *For two matrices $A, B \in \mathbb{R}^{n\times n}$, $\lambda_{\max}(A+B) \leq \lambda_{\max}(A) + \lambda_{\max}(B)$*

Proof. We can express the maximum eigenvalue of the matrix $A+B$ as

$$\lambda_{\max}(A+B) = \max_{\mathbf{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} \frac{\mathbf{x}^T(A+B)\mathbf{x}}{\|\mathbf{x}\|^2} \leq \max_{\mathbf{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} \frac{\mathbf{x}^T A\mathbf{x}}{\|\mathbf{x}\|^2} + \max_{\mathbf{x}\in\mathbb{R}^n\setminus\{\mathbf{0}\}} \frac{\mathbf{x}^T B\mathbf{x}}{\|\mathbf{x}\|^2} \tag{23}$$

$$= \lambda_{\max}(A) + \lambda_{\max}(B) \tag{24}$$

□

Finally, applying Lemma 1 $n-1$ times,

$$\lambda_{\max}(\nabla^2 f) \leq \sum_{i=1}^{n} \frac{\|\mathbf{a}_i\|^2}{2} =: \frac{\|A\|_F^2}{2} =: L \tag{25}$$

1.I.5 For $g(\mathbf{x}) := \|\mathbf{x}\|_1$,

$$\text{prox}_{\lambda g}(\mathbf{z}) := \arg\min_{\mathbf{y}\in\mathbb{R}^d} \left\{ \lambda \|\mathbf{y}\|_1 + \frac{1}{2}\|\mathbf{y}-\mathbf{z}\|_2^2 \right\} = \arg\min_{\mathbf{y}\in\mathbb{R}^d} \left\{ \sum_{i=1}^{d} \lambda|y_i| + \frac{1}{2}(y_i - z_i)^2 \right\} \tag{26}$$

So using the first order optimality condition,

$$\frac{\partial}{\partial y_j} \left\{ \sum_{i=1}^{d} \lambda|y_i| + \frac{1}{2}(y_i - z_i)^2 \right\} = \lambda s + (y_j - z_j) = 0 \tag{27}$$

where $s \in \partial|y_j|$. We can distinguish, the following cases, where s is taken to be 0 in the case $y_j = 0$:

$$y_j = \begin{cases} z_j - \lambda & \text{if } y_j > 0, \text{ so } z_j > \lambda \\ z_j + \lambda & \text{if } y_j < 0, \text{ so } z_j < -\lambda \\ 0 & \text{if } y_j = 0 \end{cases} \tag{28}$$

which can be summarised as $y_j = \max(|z_j| - \lambda, 0)\text{sign}(z_j)$. Finally, we have

$$\nabla_{\mathbf{y}} \left\{ \sum_{i=1}^{d} \lambda |y_i| + \frac{1}{2}(y_i - z_i)^2 \right\} = 0 \iff \mathbf{y} = \max(|\mathbf{z}| - \lambda, \mathbf{0}) \circ \text{sign}(\mathbf{z}) \tag{29}$$

where the operators $\max, \text{sign}$ and $|\cdot|$ are applied coordinate-wise. So the resulting proximal operator is

$$\text{prox}_{\lambda g}(\mathbf{z}) = \max(|\mathbf{z}| - \lambda, \mathbf{0}) \circ \text{sign}(\mathbf{z}) \tag{30}$$

1.I.6 For $g(\mathbf{x}) := \frac{1}{2}\|\mathbf{x}\|_2^2$,

$$\text{prox}_{\lambda g}(\mathbf{z}) := \underset{\mathbf{y} \in \mathbb{R}^d}{\arg\min} \left\{ \lambda \frac{1}{2}\|\mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y} - \mathbf{z}\|_2^2 \right\} \tag{31}$$

Again equating the gradient to zero,

$$\nabla_{\mathbf{y}} \left\{ \lambda \frac{1}{2}\|\mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y} - \mathbf{z}\|_2^2 \right\} = \lambda \mathbf{y} + (\mathbf{y} - \mathbf{z}) = 0 \iff \mathbf{y} = \frac{\mathbf{z}}{1 + \lambda} \tag{32}$$

so the proximal operator in this case is

$$\text{prox}_{\lambda g}(\mathbf{z}) = \frac{\mathbf{z}}{1 + \lambda} \tag{33}$$

Part II - Handwritten Digit Classification

Discussion of the convergence plots

As we can see in Figures 1 and 2, the convergence with ISTA algorithm is very slow for both ℓ_1 and ℓ_2 regularized logistic regression. Note that for the latter case it offers the poorest performance among all the tested algorithms.

Regarding the Stochastic Proximal method, we observe a flat behavior for the ℓ_1 regularization and a better performance than the deterministic ISTA algorithm for ℓ_2 regularization. Differently from the deterministic algorithms, the x axis of the stochastic plot is the number of epochs, so it would correspond to one iteration of a deterministic method. Hence, `PROX-SG` has effectively half as many iterations as the deterministic methods.

Note that the previous algorithms offer a very slow convergence since the step size is chosen to be the Lipschitz constant of the gradients for each of the problems. This is of the order of 10^5, hence the resulting step size is order 10^{-5}, which explains the flat behavior of the convergence curves.

For this reason, accelerated algorithms are specially suited in this case. As expected, FISTA performs better than any other algorithm for these problems and its restarted version is equivalent until the occurrence of the first ripple. By construction, the restarted version avoids the oscillatory behavior of accelerated algorithms and converges faster in the number of iterations. Nevertheless, the restarted version has a higher computational cost per iteration since it has to compute additional updates when the restart condition is satisfied. We can see that for ℓ_1 regularization (Figure 1), it's not worth incorporating the restart condition.

In the case of ℓ_2 regularization (Figure 2), we can see that the oscillatory behavior characteristic of accelerated methods is much more significant. Hence, the restarted version of FISTA would be preferred since, even if it takes more time to perform the same number of iterations, it achieves an objective value several orders of magnitude closer than without restart.

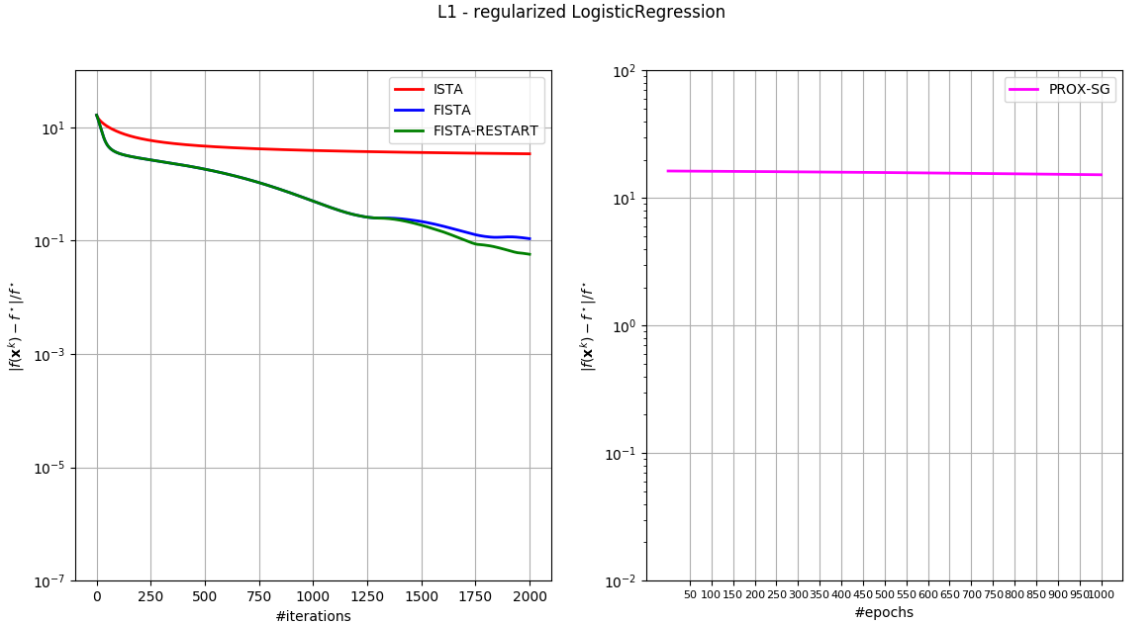

Figure 1: Convergence of ℓ_1-regularized Logistic regression

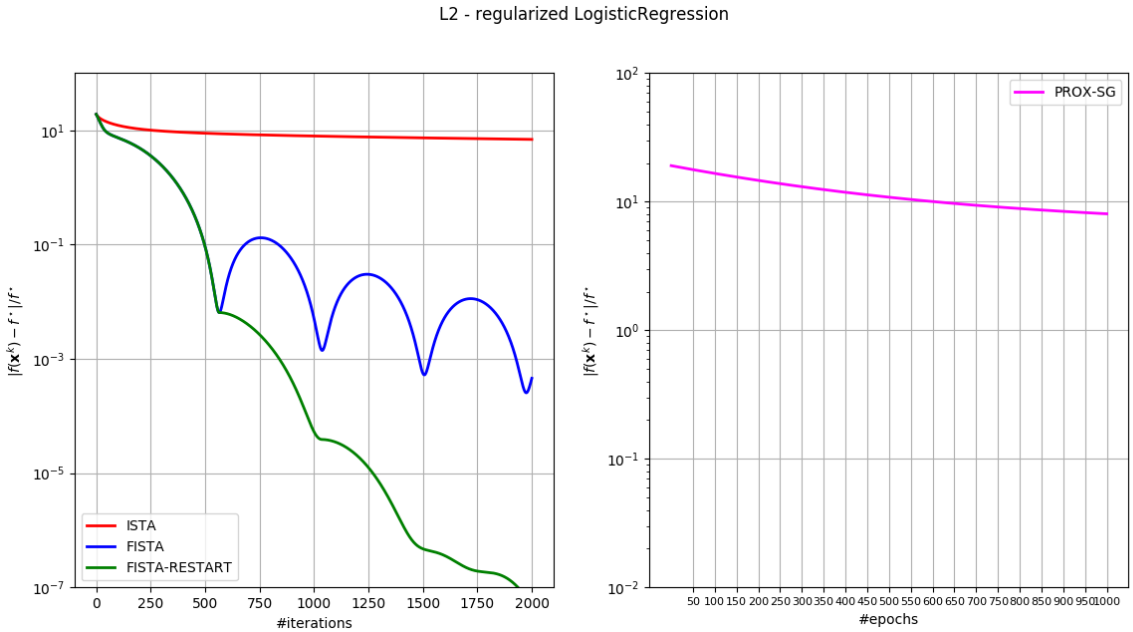

Figure 2: Convergence of ℓ_2-regularized Logistic regression

Theoretical convergence rates

As mentioned in the previous section, the slow convergence is due to the large Lipschitz constant L of the gradient of the objective function, which translates into a very little step size. The theoretical convergence

rates of both ISTA and FISTA algorithms have the form

$$\frac{F(\mathbf{x}^k) - F^*}{F^*} \leq \begin{cases} \frac{F^* L R_0^2}{2(k+2)} & \text{for ISTA algorithm} \\ \frac{2F^* L R_0^2}{(k+2)^2} & \text{for FISTA algorithm} \end{cases} \tag{34}$$

where $R_0 = \|\mathbf{x}_0 - \mathbf{x}^*\|$. These bounds are depicted as dashed lines in Figure 3. Here we can see that, even if the convergence is in general very slow as discussed in the previous section, the empirical results are below the theoretical bounds, which have a large value due to the huge L factor in (34).

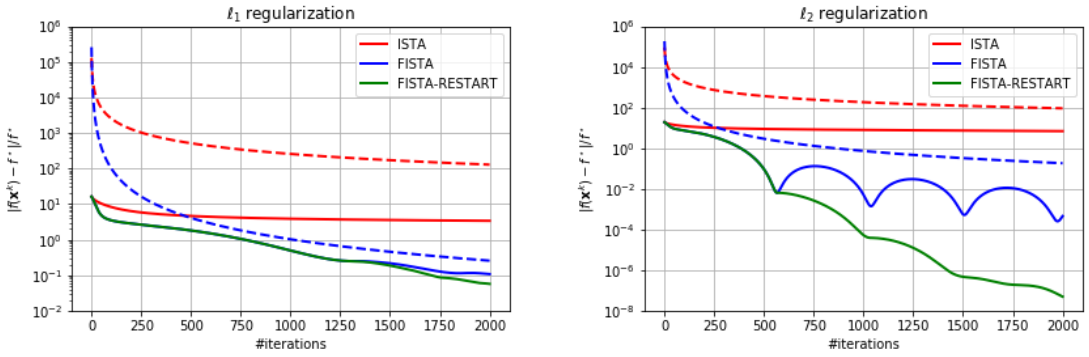

Figure 3: Comparison against theoretical convergence rates

Digit features

Figures 4 and 5 depict the feature matrices obtained by running FISTA with gradient restart. Specially for the case of ℓ_2 regularization (Figre 5), we can see similarities between each class and the depicted values. This can be easily seen for classes 0, 1 and 3.

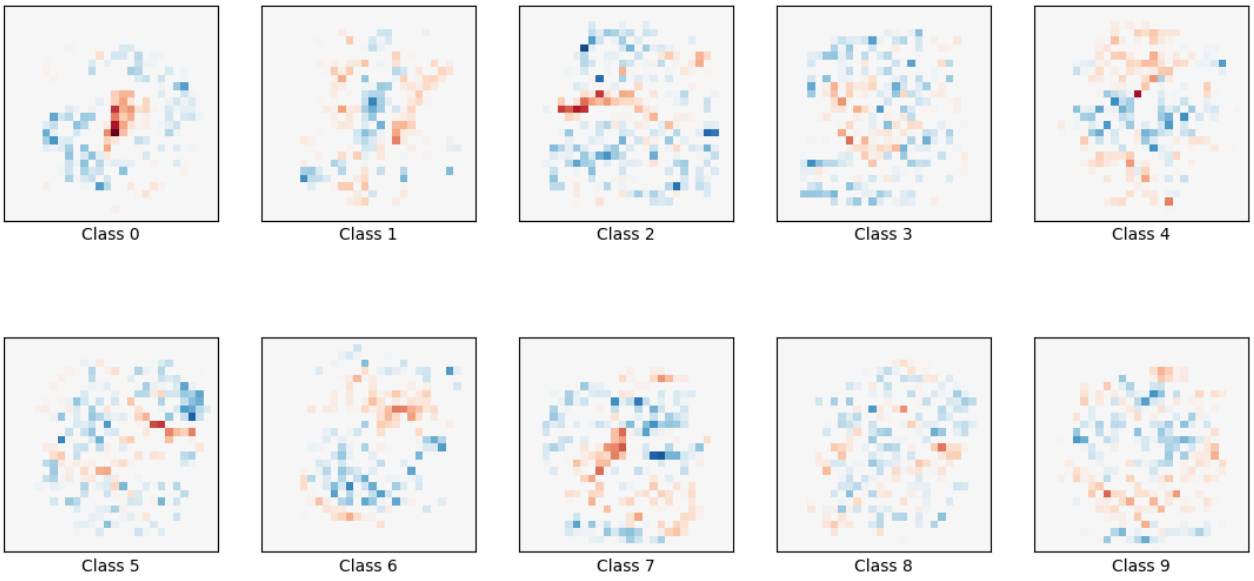

Figure 4: Visualization of the feature matrices for ℓ_1-regularized Logistic Regression

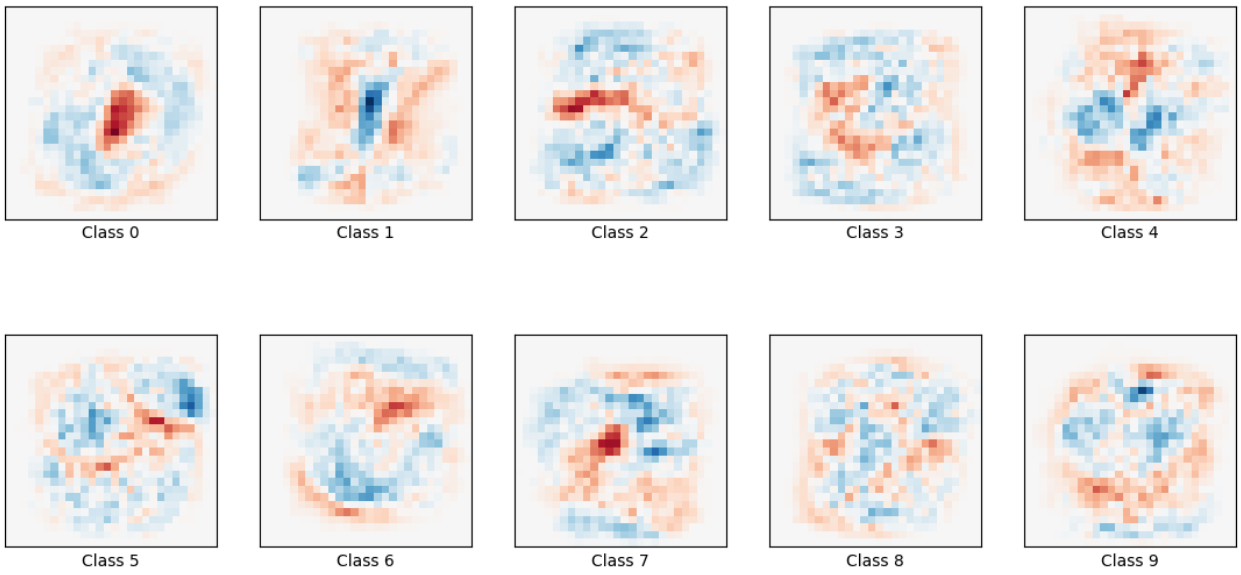

Figure 5: Visualization of the feature matrices for ℓ_2-regularized Logistic Regression

Comparison against Neural Network

The test accuracies of both regularization versions of the Logistic Regression formulation are presented in Table 1 along the test accuracy of a Neural Network. The tested model for Logistic regression is the one obtained by FISTA with gradient scheme restart for both ℓ_1 and ℓ_2 regularization, while the Neural Network is a 3-layer perceptron with ReLU activation functions.

As seen in Table 1, one can intuit that the Logistic Regression model has not as many expressive power as the Neural Network. The universal approximation theorem states that a 1-hidden-layer Neural Network with enough hidden units and a non-polynomial activation function can approximate a continuous function arbitrarily well. The theorem may not hold by the *enough hidden units* condition, the same reasoning applies. Nevertheless, in this case we don't have a wide Neural Network but a one deeper that only one layer, which potentially increments the expressive power.

On the other hand, one can understand the Logistic Regression formulation as a one-layer neural network, i.e. no hidden layers, and with softmax activation function. Note that this activation function has no discontinuities and can be arbitrarily well approximated using a Taylor series. Moreover, in this case there is no hidden layer. In conclusion, in any case the general approximation theorem holds for Logistic Regression.

ℓ_1-regularized Logistic Regression	ℓ_2-regularized Logistic Regression	Neural Network
89.20%	89.89%	94.70%

Table 1: Test accuracies for each setup

Problem 2 - Image Reconstruction

2.1 a)

$$\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) := \nabla_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \left\| \mathbf{b} - P_\Omega W^T \boldsymbol{\alpha} \right\|_2^2 \right\} = W P_\Omega^T (P_\Omega W^T \boldsymbol{\alpha} - \mathbf{b}) \tag{35}$$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) := \nabla_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{b} - P_\Omega \mathbf{x}\|_2^2 \right\} = P_\Omega^T (P_\Omega \mathbf{x} - \mathbf{b}) \tag{36}$$

b)

$$\|\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) - \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\beta})\| = \left\| W P_\Omega^T P_\Omega W^T (\boldsymbol{\alpha} - \boldsymbol{\beta}) \right\| \leq \left\| W P_\Omega^T P_\Omega W^T \right\| \|\boldsymbol{\alpha} - \boldsymbol{\beta}\| \tag{37}$$

$$:= L_{\ell_1} \|\boldsymbol{\alpha} - \boldsymbol{\beta}\| \tag{38}$$

where the inequality follows by definition of the spectral norm.

$$\|\nabla_{\mathbf{x}} f(\mathbf{x}) - \nabla_{\mathbf{x}} f(\mathbf{y})\| = \left\| P_\Omega^T P_\Omega (\mathbf{x} - \mathbf{y}) \right\| \leq \left\| P_\Omega^T P_\Omega \right\| \|\mathbf{x} - \mathbf{y}\| := L_{TV} \|\mathbf{x} - \mathbf{y}\| \tag{39}$$

A valid upper bound in both cases is $L_{\ell_1} = L_{TV} = 1$, which follows by noting that P_Ω is a projection that selects some coordinates without altering its values and the Wavelet transform defines an orthonormal basis. In summary, both the matrices $P_\Omega^T P_\Omega$ and $W P_\Omega^T P_\Omega W^T$ will be diagonal with entries 1 in those coordinates selected by P_Ω and 0 otherwise. Hence, it's trivial to see that the maximum eigenvalue (and thus the spectral norm of these matrices) will be 1 in both cases.

2.2 A parameter sweep over λ_1 and λ_{TV} is depicted in Figure 6 against the Peak signal-to-noise ratio (PSNR) of the reconstructed image taking the original one as a reference. Each reconstruction has been obtained by running 200 iterations of FISTA. The higher the PSNR the better, and this can be seen in Figure 7.

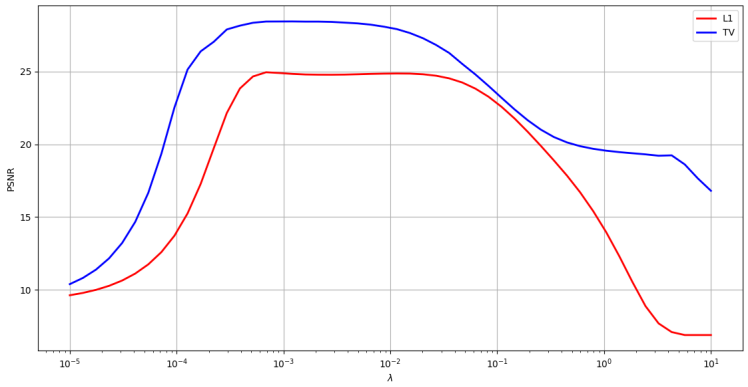

Figure 6: PSNR as a function of λ, the penalty parameter

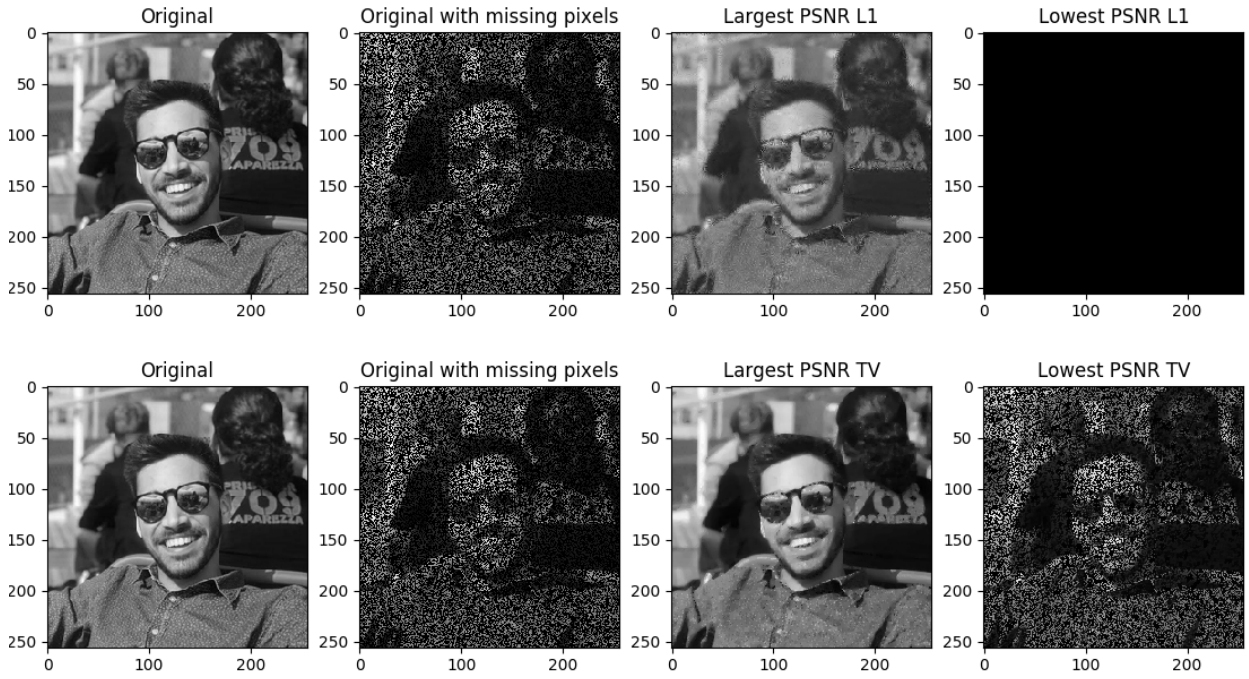

Figure 7: Reconstructed images for ℓ_1 and TV regularization with best and worst penalty

2.3 When analyzing the convergence for ℓ_1 regularization, we can see that again the FISTA with restart is the best method among the tested ones. The evolution towards the solution of this problem is depicted in Figure 8. Note that the solution to the problem formulation is not necessarily the same as the original image without missing pixels. Indeed, this can be seen in 9. Given that F evaluated at the original image is very large and for each iteration we are decreasing the value of F, we are moving away. This explains the plot with increasing relative absolute error of Figure 9.

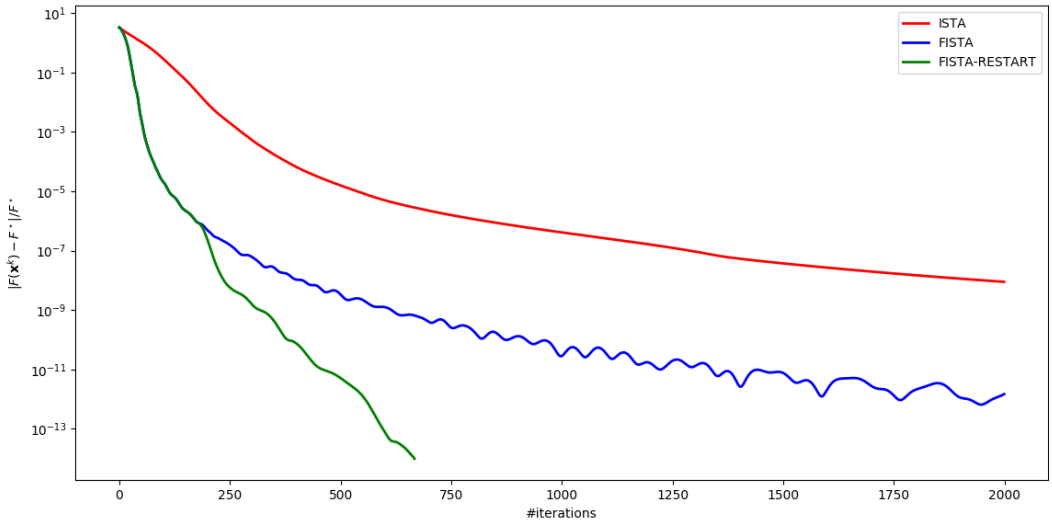

Figure 8: Relative absolute error against $F^* = 28.18$

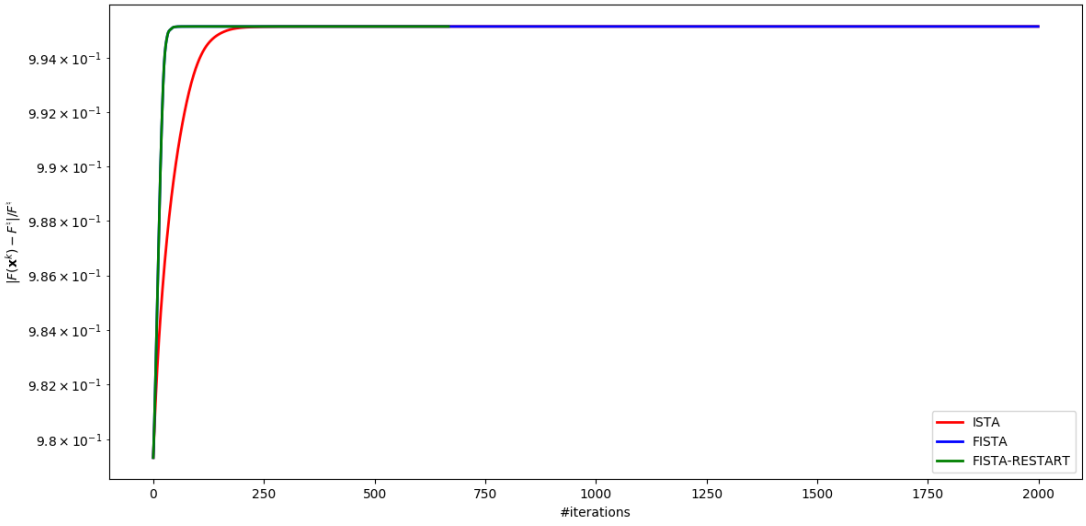

Figure 9: Relative absolute error against $F^{\natural} = 5835.05$

2.4 We can see the reconstruction obtained for both ℓ_1 and TV regularization after 500 steps in Figure 10, where apart from the original image and the original with missing pixels, an error map of the reconstruction is depicted. In Figure 11, the same results but for the unrolled proximal operator can be seen. Note that comparing the regularized methods, TV regularization obtains a better result in terms of the PSNR, the SSIM and also the time. Nevertheless, when compared with the neural network, even if the SSIM is the same, this latter has a slightly larger PSNR and the time to obtain the reconstruction is much lower. Nevertheless, the time accounts for the inference and not for the computation of the

unrolled proximal model.

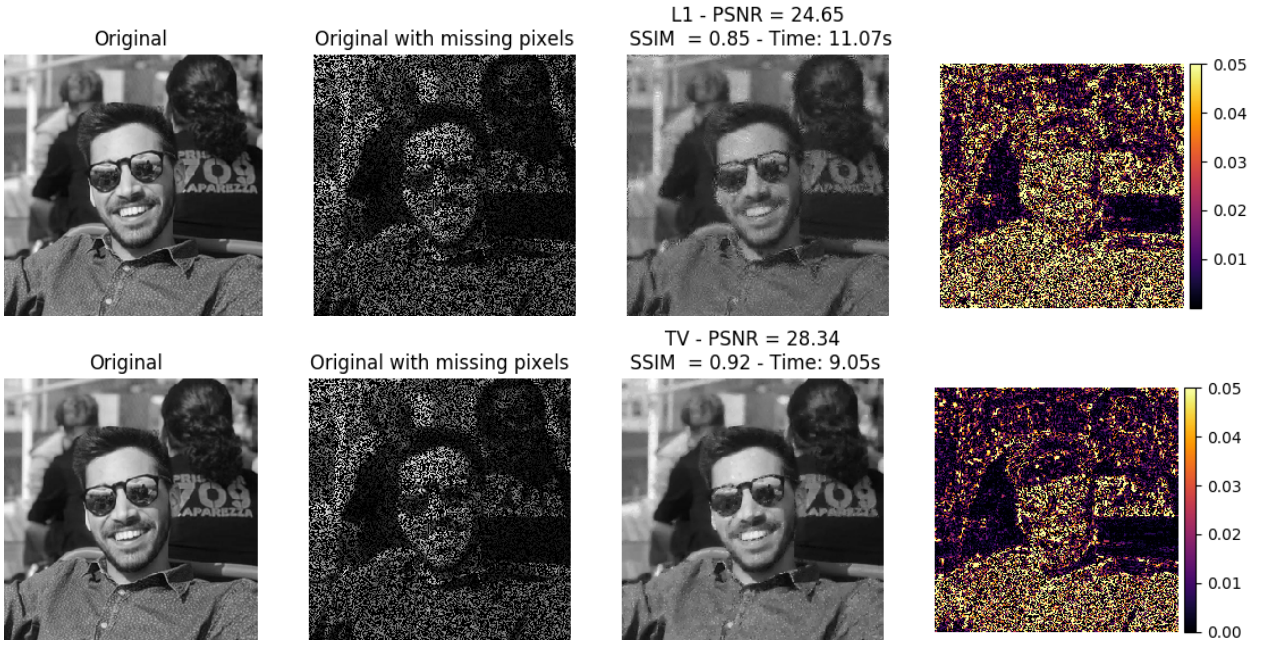

Figure 10: Reconstruction with ℓ_1 and TV-regularization and 500 iterations

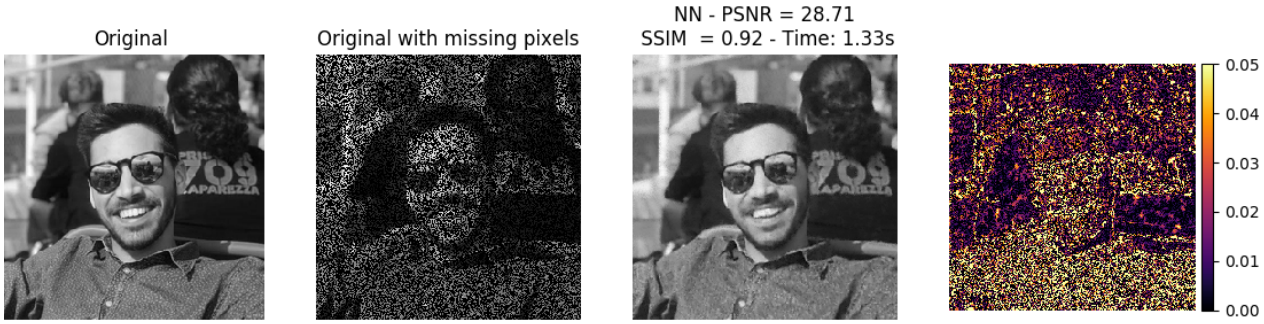

Figure 11: Reconstruction using the trained unrolled proximal operator

For a fair comparison, Figure 12 shows the results with only 5 iterations for the three methods considered before. Note that each iteration of the Neural Network costs roughly 200 times more than with the least squares problem formulation with either ℓ_1 or TV regularization. These achieve very similar performance for only 5 iterations and are comparable with the trained unrolled proximal operator with 500 iterations, which still take less than half the time needed to perform five iterations of the Neural Network. Therefore, one would probably prefer TV regularization as the in-painting method for the minor improvements with the Neural Network and the time it takes to fully train this model.

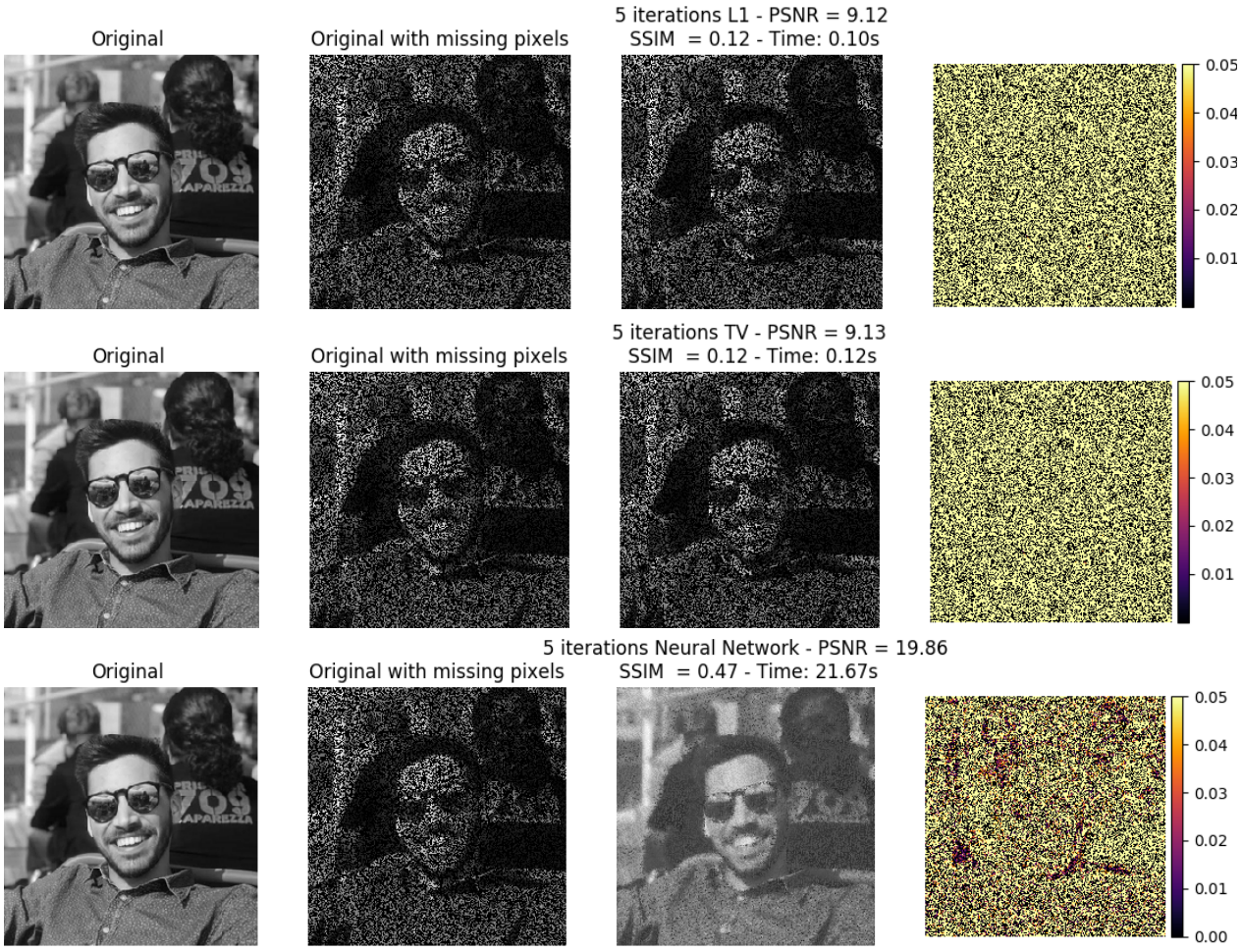

Figure 12: Reconstructions using 5 iterations