

# DeepMP for Non–Negative Sparse Decomposition

Konstantinos A. Voulgaris   Mike E. Davies   Mehrdad Yaghoobi  
Institute for Digital Communications, the University of Edinburgh, EH9 3JL, UK  
{Konstantinos.Voulgaris, Mike.Davies, m.yaghoobi-vaighan}@ed.ac.uk

**Abstract**—Non–negative signals form an important class of sparse signals. Many algorithms have already been proposed to recover such non–negative representations, where greedy and convex relaxed algorithms are among the most popular methods. The greedy techniques are low computational cost algorithms, which have also been modified to incorporate the non–negativity of the representations. One such modification has been proposed for Matching Pursuit (MP) based algorithms, which first chooses positive coefficients and uses a non–negative optimisation technique that guarantees the non–negativity of the coefficients. The performance of greedy algorithms, like all non–exhaustive search methods, suffer from high coherence with the linear generative model, called the dictionary. We here first reformulate the non–negative matching pursuit algorithm in the form of a deep neural network. We then show that the proposed model after training yields a significant improvement in terms of exact recovery performance, compared to other non–trained greedy algorithms, while keeping the complexity low.

**Index Terms:** Matching Pursuit, Non–negative Sparse Approximations, , Multilabel Classification, Deep Neural Networks

## I. INTRODUCTION

Sparse coding is the problem of reconstructing input vectors using a linear combination of an overcomplete family basis vectors with sparse coefficients. It has become extremely popular for extracting features from raw data, particularly when the dictionary of basis vectors is learned from unlabeled data. There exist several unsupervised learning methods that have been proposed to learn the dictionary. Applications of sparse coding may be found in fields such as visual neuroscience [1], [2] and image restoration [3], [4]. A major problem with these methodologies is that the inference algorithm is somewhat expensive, prohibiting real–time applications.

Let the signal of interest be  $\mathbf{y} \in \mathbb{R}^M$  and a dictionary of elements  $\Phi \in \mathbb{R}^{M \times N}$  be given. The linear sparse approximation can be formulated as finding the sparsest  $\mathbf{x} \in \mathbb{R}^N$ ,  $M < N$ , i.e having the minimum number of non–zero elements, as follows:

$$\mathbf{y} \approx \Phi \mathbf{x} \quad (1)$$

The greedy sparse approximation algorithms are in general known for low computational costs, suitable for real–time and large scale sparse approximations. The Matching Pursuit (MP) [5], algorithm is introduced, which approximately solve the following problem:

$$\tilde{\mathbf{x}} := \operatorname{argmin}_{\mathbf{x}_s} \|\mathbf{y} - \Phi_s \mathbf{x}_s\|_2, \quad (2)$$

where  $s$  is a subset of all atoms with cardinality  $k$ .

There are many applications for which the coefficient vectors are not only sparse, but also non–negative. Spectral and multi–spectral unmixing, [6], [7], microarray analysis [8] and Raman spectral deconvolution [9] are a few examples.

The original implementation of MP has been modified in order to adopt the algorithm to the non–negativity setting. Essentially the original minimization problem introduced in (2) is reformulated by adding a constraint that guarantees the non–negativity of the coefficients and takes the following form:

$$\tilde{\mathbf{x}} := \operatorname{argmin}_{\mathbf{x}_{\geq 0}} \|\mathbf{y} - \Phi \mathbf{x}\|_2 \quad (3)$$
$$\|\mathbf{x}\|_0 \leq j$$

where  $\|\cdot\|_0$  measures the number of non–zero elements.

MP incrementally builds up  $\mathbf{x}$  with respect to the set of columns of  $\Phi$ ,  $s_a$ . A known fact about MP algorithms is that the obtained solution can be an approximation of the input signal  $\mathbf{y}$ . The acquired support set  $s_a = \{i : \phi_i \in \Phi\}$  is then different to the ground truth support set  $s_g$ . Hence, given a  $k$ –sparse signal  $\mathbf{x}$ , it is quite frequent to have  $|s_a \cap s_g| < k$ , particularly when the atoms in the dictionary are correlated, i.e coherent dictionary.

The authors in [10] introduced the Learned Iterative and Thresholding Algorithm (LISTA). Under the assumption that the basis vectors have been trained and are being fixed, the core idea is to train a parameterized encoder function to predict the optimal sparse code. A key advantage of this physical–model based framework is that it has a predetermined–complexity and can be used to approximate sparse codes with a fixed computational cost and a prescribed expected error that makes it appealing for real–time applications which is the main focus of this study. Recent advances in the LISTA framework [11] introduced the theoretical conditions upon the convergence of the algorithm, while in [12] the authors demonstrated that by following an analytical approach, rather than a learned approach, the network retains its optimal linear convergence. They later introduce an acceleration technique in the training procedure, given that the number of training parameters is significantly reduced.

Inspired by the unfolding idea introduced by the LISTA framework to reformulate the convex optimization algorithms with deep neural networks (DNN), we here introduce a variation of the original NNMP, as introduced in **Algorithm 1**, called DeepMP as a data adaptive and bounded complexity algorithm. Our preliminary goal is to introduce a novel framework for non–negative sparse approximation which outper-

```

1: initialisation:  $s = \emptyset$ ,  $k = 0$  and  $\mathbf{r}_0 = \mathbf{y}$ 
2: while  $k < K$  &  $\max(\Phi^T \mathbf{r}_k) > 0$  do
3:    $\mathbf{s}_k = \mathbf{0}$ 
4:    $(\zeta, \ell) \leftarrow \max(\Phi^T \mathbf{r}_k)$ 
5:    $\mathbf{s}_k[\ell] = \zeta$ 
6:    $\mathbf{r}_{k+1} \leftarrow \mathcal{P}\{\mathbf{r}_k - \zeta \phi_\ell\}$ 
7:    $k \leftarrow k + 1$ 
8: end while
9:  $\mathbf{x} \leftarrow \sum_k \mathbf{s}_k$ 

```

**Algorithm 1:** Non-Negative Matching Pursuit algorithm (NNMP)

forms the existing ones in terms of accuracy, maintaining the computational cost. Nevertheless the canonical linear approach followed at the selection step of NNMP even though it is simple computationally is not data adaptive and therefore not flexible. The DeepMP approach has more flexible nature introducing a higher degree of freedom at the selection step representation underlying complex nature of data i.e human handwriting.

## II. DEEP MATCHING PURSUIT

In this section we introduce the modification in the standard NNMP structure which is introduced in Algorithm 1. Starting with the measurement  $\mathbf{y}$  as the current residual signal  $\mathbf{r}_k|_{k=0}$ , the main steps are: a) finding the best matched atom  $\phi_k$  to  $\mathbf{r}_k$ , and 2) updating the residual  $\mathbf{r}_k$  by subtracting the contribution of selected atom. The operator  $\mathcal{P}$  can be identity or the projection onto the positive orthant which is done such that the framework is consistent with the non-negative setting. A flowchart diagram of each NNMP iteration has been shown in Figure 1, where "max" operator simply keep the largest component of the input and zero out the rest, and  $\mathbf{s}_k$  is the 1-sparse vector with the appropriate coefficient on its support. The "max" operator is called "hard-max" operation here, which is the projection onto the best one-sparse set, also known as the 1-sparse hard-thresholding [13].

The NNMP steps are reformulated and the dictionary is replaced with the weight matrices of the same size  $\mathbf{W}_f^{(k)}$  and  $\mathbf{W}_b^{(k)}$  at iteration  $k$ . We then get the following (non-linear) system of equations,

$$\begin{pmatrix} \mathbf{r}_{k+1} \\ \mathbf{x}_{k+1} \end{pmatrix} = g_b \begin{pmatrix} -\mathbf{W}_b^{(k)} g_f(\mathbf{W}_f^{(k)T} \mathbf{r}_k) + \mathbf{r}_k \\ g_f(\mathbf{W}_f^{(k)T} \mathbf{r}_k) + \mathbf{x}_k \end{pmatrix}, \quad (4)$$

where  $g_f$  and  $g_b$  are respectively forward and backward functions which are hard-max and  $\mathcal{P}$ . Such a model can be represented as two layers of a neural network model per single iteration of the algorithm. By concatenation of  $K$  blocks of Figure 2. The depth of the network then varies depending on the sparsity of the signal  $k$ , and can be represented as the concatenation of  $k$  blocks of Figure 3. The network takes then the form of a DNN of  $2K$  layers.  $\mathbf{x}$  can then be reconstructed by superposition of  $\mathbf{s}_k \mathbf{s}$ .

This structure provides a framework for sparse approximations, if we train the weight matrices using the backpropaga-

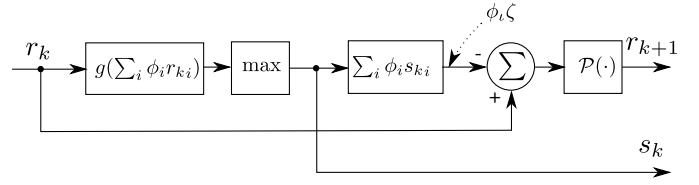


Fig. 1: One step of non-negative matching pursuit algorithm.  $r_k$  and  $s_k$  are respectively the residual and the selected index at the  $k^{th}$  step of algorithm.

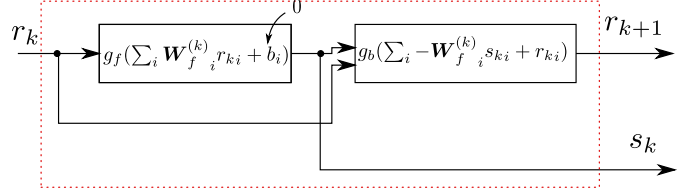


Fig. 2: Representation of one step of NNMP algorithm in the form of a two layer neural network with a skip connection.  $g_f$  and  $g_b$  are respectively hard-max and linear/ReLU activation functions.

tion algorithm. Such a variational learning method with non-differentiable activations is possible using a surrogate function optimization method, e.g. [14] for ReLU. As one candidate solution is  $\mathbf{W}_f^{(k)} = \mathbf{W}_b^{(k)} = \Phi, \forall k$ , the network will at least work as good as NNMP, if the learning is successful. Within our work we modify step 4 of NNMP by replacing the original library  $\Phi$  with  $\mathbf{W}_b^{(k)}$  while in step 6  $\mathbf{W}_f^{(k)} = \Phi$ . At the particular step we introduce the non-linear activation function  $\mathcal{P}$  with the aim to project the residual vector  $\mathbf{r}$  on the positive orthant.

Essentially the key change in the NNMP structure is the more flexible approach in step 4 of the algorithm. A successful decomposition of the input signal  $\mathbf{y}$  depends on the proper selection of the candidate atom. A wrong selection can have a direct impact on the minimization problem introduced in (2). By training DeepMP we generate different copies of  $\mathbf{W}_b^{(k)}$  over the  $k$  layers of the network. We aim to have an approach that prevents misclassifications and potentially results in zero error for a fixed number of iterations and no noise in the input. Our expectation relies on the fact that DeepMP utilizes a higher degree of freedom in the selection step of the algorithm. In particular, the canonical approach of NNMP represents a model that consists of a fixed number of  $MN$  parameters. This kind of approach may be simple, but not flexible. On the other hand the DeepMP model consists of a number of parameters which scales linearly over the network layers and results in the total to a number of  $KMN$  parameters at the selection step. In practice this means that the higher the sparsity of the signal  $K$  the higher degree of flexibility introduced in the framework, i.e the capacity of the DNN, which improves the performance compared to canonical NNMP overall.

From classification point of view the DeepMP framework actually performs a multilabel classification task by decomposing the input signal  $\mathbf{y}$  with respect to the corresponding

classes, and using the categorical cross-entropy loss function,

$$H(p, q) = - \sum_{j=1}^{|\Phi|} \mathbf{1}_{s_a}(j) \log p(j, i) \quad (5)$$

where  $i$  corresponds to the  $i$ -th sample,  $j$  to the index of the atom and  $\mathbf{1}_{s_a} : I \rightarrow \{0, 1\}$  is the indicator function, defined as:

$$\mathbf{1}_{s_a}(j) = \begin{cases} 1 & \text{if } j \in s_a \\ 0 & \text{if } j \notin s_a \end{cases}$$

#### A. Sparse Signal Decomposition

The main motivation for introducing deep learning approach is to introduce more flexibility approach in the selection rule of the MP type algorithms. We aim to improve the prediction rate on the support set and eventually reduce the residual error compared to the standard MP framework.

Considering the set of sparse signals which are the main focus of the current work. The main goal of the decomposition algorithm is to identify the atoms which build up the input signal  $\mathbf{y}$  with non-negative weights as follows:

$$\mathbf{y} = \sum_{l=1}^k a_w \phi_i. \quad (6)$$

with  $a_w \sim U[0, 1]$ , where  $U[0, 1]$  stands for the uniform distribution with 0 mean and unit variance.

The overall process can then be represented as an iterative algorithm. A common phenomenon that frequently takes place during the decomposition is the selection of unrelated atoms in the support set  $s_a$ , over the iterations of the algorithm.

A reason for this phenomenon occurs, is related to the similarity between the atoms  $\phi_i$ . In cases where the algorithm operates over a point cloud where the constituent atoms are highly coherent with each other, the algorithm may select a neighboring atom instead of the ground truth atoms in  $s_g$ , i.e  $i \in s_a$  while  $i \notin s_g$ . Coherence measures the maximum similarity between two distinctive atoms of  $\Phi$ . Given a pair of points  $\phi_i, \phi_j \in \Phi$  where  $i \neq j$ , the coherence can be formulated as follows:

$$\mu(\Phi) = \max_{i \neq j} \frac{|\langle \phi_i, \phi_j \rangle|}{\|\phi_i\|_2 \cdot \|\phi_j\|_2} \quad (7)$$

where  $\|\cdot\|_2$  indicates the Euclidean norm. By introducing the matrix  $\mathbf{W}_b^{(k)}$  at the selection step of the algorithm, we are aiming for the points to be represented in a way that the mutual coherence of the points will decrease. In that sense by training the network we are expecting that the coherence of the corresponding representation  $\mathbf{W}_b^{(k)}$  yields an outcome where ideally  $\mu(\Phi) \geq \mu(\mathbf{W}_b^{(k)})$ , where  $\mu(\Phi)$  and  $\mu(\mathbf{W}_b^{(k)})$  are respectively the coherence in  $\Phi$  and  $\mathbf{W}_b^{(k)}$ .

|                | N   | M    | lr     | final_lr | epochs |
|----------------|-----|------|--------|----------|--------|
| Synthetic data | 30  | 200  | $1e-3$ | 0.1      | 20     |
| Raman library  | 503 | 2521 | $1e-3$ | 0.1      | 30     |

TABLE I

### III. EXPERIMENTS

Within the current section we evaluate the performance of DeepMP by some simulations. In order to evaluate the performance of DeepMP we are considering two datasets; a synthetic dataset  $\Phi \in \mathbb{R}_+^{d \times N}$ . The dictionary was randomly generated with an *i.i.d.* normal distribution and then projected onto the positive orthant and column normalised. A real dataset of Raman spectra, where each of the spectras consists of 503 wavenumbers that lay within the range of 306 to 1249  $\text{cm}^{-1}$ , provided by [15]. We perform a number of 150000 trials for each dataset where only  $\mathbf{W}_f^{(k)}$ s were trained in the  $M$ -space while  $\mathbf{W}_b^{(k)} = \Phi$ . This essentially means that we only train the weights that correspond to the selection step of the algorithm while the weights that correspond to the update step are kept fixed. This is done because we address DeepMP as a solver for the standard non-negative least squares problem as introduced in (3). In case where the weights in the update step are also trained the cost function of the problem is reformulated as follows:  $\text{argmin}_{\mathbf{x}_s \geq 0} \|\mathbf{y} - \mathbf{W}_{b,s}(k)\mathbf{x}_s\|_2$ .

The DeepMP framework is optimized using the *AdaBound* algorithm [16]. More details about the datasets and the settings for the *AdaBound* algorithm can be found in TABLE I.

As an evaluation metric for the exact recovery of the support set we are using the normalized *Hamming distance* complement [17]. The metric is defined as in (8).

$$H^c(s_a, s_g) = \sum_{n=1}^N 1 - \frac{|s_a(n) - s_g(n)|}{k} \quad (8)$$

where  $S_a$  is the support set acquired by the corresponding algorithm and  $S_g$  the ground truth,  $k$  is the sparsity level and  $|\cdot|$  the cardinality operator. The performance on the reconstruction error for each sparsity  $k$  is evaluated with respect to  $\epsilon$  as follows:

$$\epsilon(k) = \frac{1}{Z} \sum_{z=1}^Z \frac{\|\mathbf{y}[z] - \Phi \mathbf{x}[z]\|_2}{\|\mathbf{y}[z]\|_2}, \quad (9)$$

where  $Z$  the number of realizations.

A basic expectation while training the selection step of the algorithm is the variation of the coherence in between  $\mathbf{w}_i^k, \mathbf{w}_j^k$  columns of  $\mathbf{W}_b^{(k)}$ . For the particular aspect of the problem we use the empirical cumulative distributed function (ECDF) as introduced in equation (10):

$$\mu_{\text{ECDF}}(t) = \frac{1}{\binom{|\Phi|}{2}} \sum_{i=1}^M \sum_{j \neq i}^M \mu(\phi_i, \phi_j) \leq t. \quad (10)$$

where  $t \in [0, 1]$ .

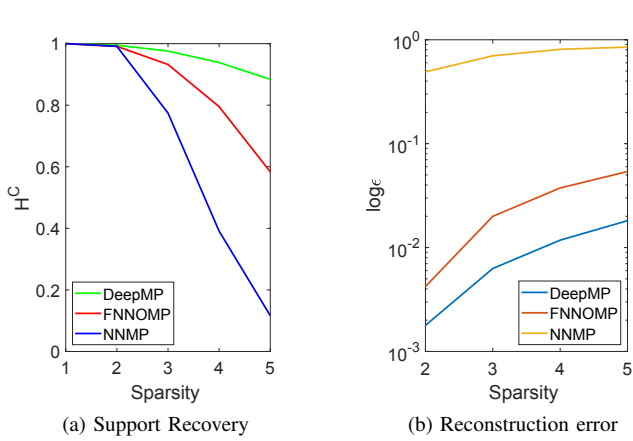


Fig. 3: The performance of different MP frameworks with the *Raman* dictionary.

### A. Results

We here perform a simulation based evaluation of the different MP frameworks. We are particularly interested in signals which are very sparse. Hence we consider mixtures of signals  $\mathbf{y}$  that consist of up to 5 atoms. From the perspective of the DeepMP framework this corresponds to concatenation of up to 5 different versions of the model with a varying depth over sparsity. These versions are independent, i.e the 1st layer is different from the one model to the other. The obtained results for the *Raman* data and the synthetic data are demonstrated in figures 3 and 4 accordingly.

As it can be seen from the results, DeepMP outperforms the NNMP and FNNOMP [18] with respect to the Hamming distance complement, while FNNOMP significantly outperforms NNMP. This essentially means that the extra degrees of freedom on the selection step of DeepMP introduces a more flexible approach which overall, leads to a better exact recovery performance. The advantage of DeepMP becomes

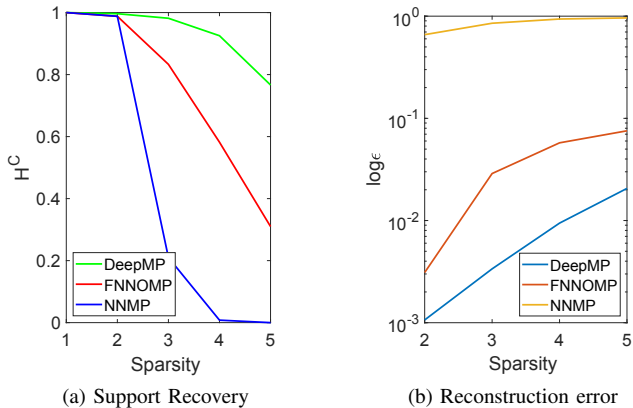
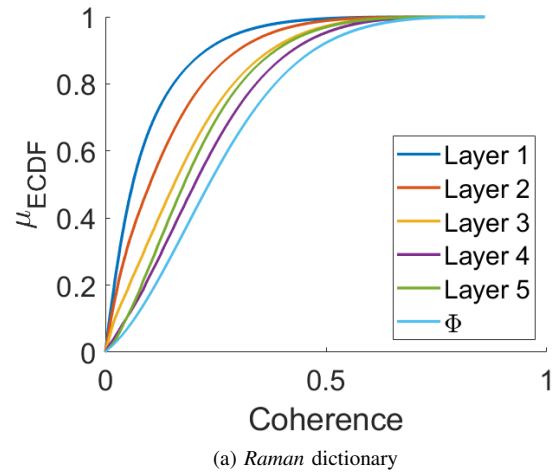
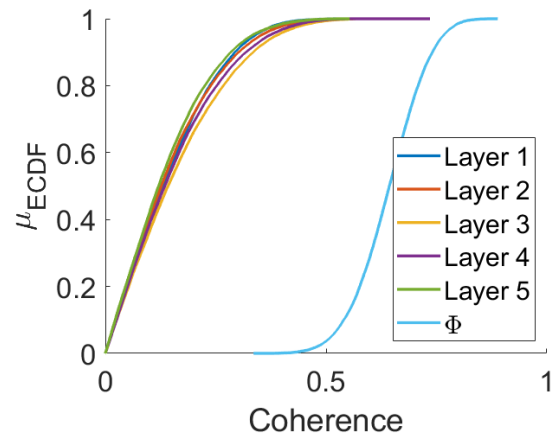


Fig. 4: The figure demonstrates the performance of the different MP frameworks for the Synthetic dictionary.



(a) *Raman* dictionary



(b) Synthetic dictionary

Fig. 5: The coherence of the *Raman* dictionary and the synthetic dictionary in comparison with the coherence of the weight matrices of DeepMP for  $k = 5$ .

more significant over sparsity having less sparse signals, i.e larger  $k$ . Hence, despite the fact that the performance of all the MP frameworks decays with  $K$ , the flexibility of DeepMP leads to a slower decay over sparsity and hence getting a better performance compared to FNNOMP and NNMP. The  $\epsilon$ -error results also indicate that the improved exact recovery, leads to a better performance on the reconstruction of the input signal  $\mathbf{y}$ .

Despite demonstrated good results using DeepMP, a question is why it outperforms NNMP and FNNOMP. While a rigorous answer to this question is left for the future, we demonstrate the  $\mu_{\text{ECDF}}$ 's of the two dictionaries and the trained model trained for  $k = 5$  in figure 5. As it can be seen from the results, the network generates weight matrices with reduced coherences compared to the original ones. In that sense, the corresponding point clouds consist of a set of atoms which are further apart the one to the other. This essentially means that DeepMP alters the underlying geometry of the selection step to avoid a misclassification. Given that the points are

further apart, i.e smaller coherence in average, the algorithm can easier pick the right atom without confusing it with its neighbors.

#### IV. CONCLUSION–FUTURE WORK

We here introduces DeepMP which is a novel framework for non–negative sparse decomposition. The main goal of the current work is to maintain the computational advantages of the standard MP algorithm while improving the performance of signal approximation. The obtained results indicate that DeepMP outperforms the standard MP approaches in terms of signal reconstruction. A future direction of the current work can be the incorporation of the matrix factorization during the training process to boost the performance of DeepMP.

#### V. ACKNOWLEDGEMENT

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant numbers EP/S000631/1 and EP/K014277/1 and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

#### REFERENCES

- [1] B.A Olshausen and D. Field “Emergence of simple–cell receptive field properties by learning a sparse code for natural images“. *Nature*,381 (6583):607–609,1996.
- [2] P.O Hoyer “Non–Negative matrix factorization with sparseness constraints“. *JMLR*,5:1457–1469,2004.
- [3] M. Elad and M. Aharon “Image denoising via learned dictionaries and sparse representation“.In *CVPR* ,2006.
- [4] M. Ranzato, F–J Boureau, L Y and Y. LeCun “Unsupervised learning of invariant feature hierarchies with applications to object recognition“.In *CVPR* ,2007.
- [5] S Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,“*IEEE Trans. on Pattern Analysis and Machine Intelligence* , vol. 11, pp.647–693,1989.
- [6] MD Iordache, JM Bioucas-Dias, and A Plaza, “Sparse unmixing of hyperspectral data,“ *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 6, pp. 2014–2039, 2011.
- [7] Y Qian, S Jia, J Zhou, and A Robles-Kelly, “Hyperspectral unmixing via sparsity-constrained nonnegative matrix factorization,“ *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 11, pp. 4282–4297, 2011.
- [8] H Kim and H Park, “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,“ *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.
- [9] D Wu, M Yaghoobi, S. I Kelly, M. E Davies, and R Clewes, “A sparse regularized model for raman spectral analysis,“ in *Sensor Signal Processing for Defence*, Edinburgh, 2014.
- [10] K. Gregor and Y. LeCun, “Learning Fast Approximations of sparse coding “, in *Proceedings of the 27th International Conference on Machine Learning*. Omnipress, 2010, pp. 399–406.
- [11] Chen, Xiaohan, et al, “Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds“. *Advances in Neural Information Processing Systems*,2018.
- [12] J. Liu, X.Chen, Z. Wang, W.Yin, “ALISTA: Analytic weights are as good as learend weights in LISTA“. *international Conference on Learning Representations (ICLR)* ,2019.
- [13] T. Blumensath and M. Davies, “Iterative Thresholding for Sparse Approximations“, *Journal of Fourier Analysis and Applications*, vol.14 no.5 pp.629-654,2008.
- [14] V Nair and G.E Hinton “Rectified linear units improve restricted boltzman machines“, in *Proceedings on the 27th international conference on machine learning(ICML-10)*, 2010,pp. 807-814.
- [15] <https://www.stjapan.de/>
- [16] L. Luo, Y. Xiong, Y. Liu, X. Sun, “Adaptive Gradient Method With Dynamic Bound of Learning Rate“, *ICLR*,2019.
- [17] A. Bookstein, V. Kulyukin, T. Raita, “Generalised Hamming Distance“, *Springer*,2002.
- [18] M Yaghoobi, D. Wu, and M. Davies, “Fast Non–Negative Orthogonal Matching Pursuit“, *IEEE Signal Processing Letters*, vol.22, no.9,2015.
- [19] T. T. Nguyen, J. Idier, C. Soussen, and E.-H. Djermoune, “Non-negative orthogonal greedy algorithms, *IEEE Transactions on Signal Processing*“, vol. 67, no. 21, pp. 5643-5658.