Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence

An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Sparse Connectivity and Activity Using Sequential Feature Selection in Supervised Learning

Fariba Nasiriyan & Hassan Khotanlou

To cite this article: Fariba Nasiriyan & Hassan Khotanlou (2018) Sparse Connectivity and Activity Using Sequential Feature Selection in Supervised Learning, Applied Artificial Intelligence, 32:6, 568-581, DOI: 10.1080/08839514.2018.1486131

To link to this article: https://doi.org/10.1080/08839514.2018.1486131

Published online: 18 Jul 2018.

APPLIED ARTIFICIAL INTELLIGENCE
2018, VOL. 32, NO. 6, 568–581
https://doi.org/10.1080/08839514.2018.1486131

Taylor & Francis
Taylor & Francis Group

Sparse Connectivity and Activity Using Sequential Feature Selection in Supervised Learning

Fariba Nasiriyan and Hassan Khotanlou

Department of Computer Engineering, Bu-Ali Sina University, Hamedan, Iran

ABSTRACT

Generally, in neural networks the sparseness is a suitable regularizer in a lot of applications. In this paper, sparse connectivity and sparse representation are used to enhance solutions to the problem of classification. Sequential feature selection is then leveraged to remove redundant features and select relevant ones. Sparseness-enforcing projection operator is used to discovering the most similar vector with a predefined sparseness degree for any input vector as well. As it will be argued, the mentioned operator is approximately differentiable at every point. From the facts it is clear that the sparseness enforcing projection would be appropriate for use as a transfer function in the proposed neural network and the network can be tuned using gradient based methods. Meanwhile, an intelligent method was used to build the architecture of the proposed neural network to achieve better performance. The MNIST dataset which consists of 70,000 handwritten digits was used to train and test the method and 99.18% accuracy was achieved by classifying this dataset.

Introduction

One of the current, most significant discussions in neural networks is sparseness which was introduced by Laughlin and Sejnowski (2003). This subject claims that every neuron has only limited connections with others. Interestingly, sparse connectivity and sparse activity of neurons that are employed in a variety of machine learning algorithms have been discovered in mammalian brains (Hubel et al. 1962). To be more specific, important and notable results were achieved while implementing the sparse coding model of Olshausen and Field (2004), which gives small pieces of images of the natural scenes, and their model is capable of generating Gabor-like filters similar to features of animates primary visual cortex cells (Vinje and Gallant 2000). It is also worth mentioning that the proposed method by LeCun et al. which can remove synaptic connections in neural networks using limited connections among neurons is another implementation of sparse connectivity (Gregor and LeCun 2010).

CONTACT Hassan Khotanlou ✉ khotanlou@basu.ac.ir Bu-Ali Sina Uniersity, Fahmideh Ave., Hamedan, Iran.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

The sparseness can be explained briefly in two concepts:

- First: the sparse connectivity, which means some of the neurons in the network are active at any time (Olshausen and Field 2004).
- Second: the sparse activity, which means that every neuron in the network has only limited connections with others (not all the neurons) (Olshausen and Field 1997).

Both of the previously mentioned concepts exist and are observed in human and animal brains (Hubel and Wiesel 1962; Markram et al. 1997; Mason, Nicoll, and Stratford 1991).

To implement sparseness some approaches exist, the first one is the *L*0 pseudo norm which is a method of quantifying sparseness using the amount of non-vanishing records in a vector (Rehn and Sommer 2007). By the poor analytical properties of *L*0 pseudo norm, the Manhattan norm of the vector, which is a convex relaxation of the preceding problem, is employed instead and used in a variety of applications (Donoho 2006). Concomitantly, finding the optimal solution using *L*0 pseudo norm, results in an NP-Hard (Non-Deterministic Polynomial-Time Hard) problem (Natarajan 1995). The disadvantage of this method is that it is not scale invariant. Another method to provide this idea using the ratio of Manhattan norm and Euclidean norm of a vector was introduced by Hoyer in (2004). The Hoyer's method consists of a mathematical computation for sparseness measure which is scale invariant and will be described in detail further on.

In connection with pattern recognition and machine learning problems, which are affected strongly by selecting suitable features, the feature selection methods would be an important step. These methods generally identify subsets of data that are pertinent to a parameter called Maximum Relevance (Auffarth, López, and Cerquides 2010).

Minimum redundancy maximum relevance (mRMR) was first demonstrated by Ferri et al. (1994). It is one of the appropriate ways of feature selection generally described in its match up with relevant feature selection. It is an algorithm usually employed in methods to determine the properties of genes and phenotypes and removes their connection and relation as much as possible. It is demonstrated that the mRMR algorithm's goal is to remove redundant subsets of features. This aim helps in solving a variety of problems such as cancer diagnosis and speech recognition.

Classification of handwritten digits is a common problem and there are lots of solutions for it. The MNIST dataset which consists of 70,000 samples of handwritten digits is a famous dataset for these types of classification problems (LeCun et al. 1998). In the present study, a new method is suggested based on sparse connection and removing redundant features using sequential forward feature selection (Zongker and Jain 1996) in the hidden layer of a two-layer neural network and with sparseness enforcing projection operator (Hoyer 2004) as a transfer function. Also, performing classification by use of the extracted features by a one layer neural network in high-dimension space has been tested. The suggested method can be used for other classification problems as well. Very deep and large neural networks hold a variety of adjustable and adaptable weights. Commonly, these networks are capable of yielding errors as low as 0.35% in conjunction with flexible and affine deformation (Ciresan et al. 2010).

Proposed method

In this section, we initially explain some definitions and preparation information about the Hoyer sparseness measure method, the algorithm used to implement sparseness enforcing projection and the feature selection method that is used in the proposed method. Finally, the architecture of the proposed neural network will be presented.

Sparseness measurement approach

One of the prominent methods to achieve sparseness measure concerning the ratio of Manhattan norm and Euclidean norm of a vector was introduced by Hoyer in (2004) and consists of the computation in Equation (1):

$$\sigma : R^n \backslash \rightarrow [0,1], x \rightarrow \frac{\sqrt{n} - ||x_1/||x_2}{\sqrt{n} - 1}, \tag{1}$$

Considering $||x||_2 \leq ||x||_1 \leq \sqrt{n}||x||_2$ for all $x \in R^n$ it is obvious that σ is well defined in the above term and also sparse vectors grow as a result of selecting higher values (Laub 2005) and is scale invariant because $\sigma(\alpha x) = \sigma(x)$ for all $\sigma \neq 0$ and all $x \in R^n$. This sparseness measure supplies all touchstones introduced by Hurley and Rickard (2009) except the one which presents and supplies the fact that the sparseness of a vector is essential to be identical to the amount of sparseness in the vector which is built by multiple concatenating of the original vector; but fortunately for a correct sparseness measure this feature is not critical (Thom and Palm 2013).

In the proposed method, a sparseness enforcing projection operator was introduced by Hoyer (2004) which is appropriate for projected gradient decent means for improving and optimizing concerning σ for a predefined and ideal degree of sparseness $\sigma^* \in (0,1)$. Hoyer's solution answers the problem by discovering the closest vector with a pre-defined scale of sparseness σ^* for the absolute vector and is presented as a Euclidean projection on parameterization of the sets as Equation (2):

$$S^{(\lambda_1,\lambda_2)} : \{s \in R^n||s||_1 = \lambda_1 \wedge ||s||_2 = \lambda_2 \wedge S_{\geq 0}^{(\lambda_1,\lambda_2)} : S^{(\lambda_1,\lambda_2)} \cap R^n_{\geq 0}. \tag{2}$$

In Equation (2), it is useful that the first set makes unrestricted projection achievable. In addition, the latter set gives the point in non-negative solution and states. Strictly speaking, $\lambda 1, \lambda 2 > 0$ are target norms and it is accessible to be chosen such that all points in this set achieve the σ^* sparseness.

To perform the explained projection of Hoyer's original method, cyclic projections should be carried out that consist of projection onto a hyper plane representing the $L1$ norm constraint, a hyper sphere representing the $L2$ norm constraint, and the non-negative orthant. In another major study, Theis et al. have shown that a simpler and slightly improved version of the Hoyer's base solution, proved to be correct (Theis, Stadlthanner, and Tanaka 2005).

Algorithm for applying sparsity enforcing projection operator

For computing the sparsity enforcing projection operator we used the proposed algorithm by Thom and Palm (2013). First we consider some preparation information. Let $H := \{a \in R^n | e^T a = \lambda_1$ be the target hyper plane where all points coordinates are added together to λ_1 and $e_1, e_2, \ldots, e_n \in R^n$ be the canonical basis of the n-dimensional Euclidean space R^n. The vector H in the non-negative orthant $R^n_{\geq 0}$ is equal to the $L1$ norm constraint. Further let $K := \left\{q \in R^n \middle| \|q\|_2 = \lambda_2\right\}$ be the target hyper sphere of all points satisfying the $L2$ norm constraint. Now with the above definition for H and K we have Equation (3) (Thom and Palm 2013):

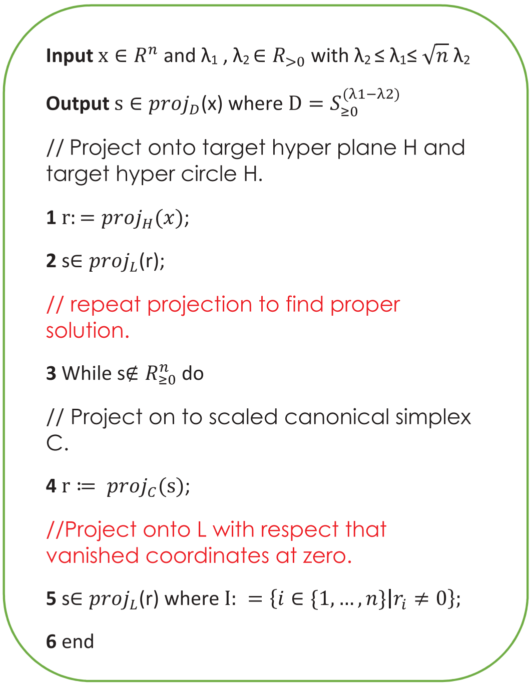
$$S^{(\lambda_1, \lambda_2)}_{\geq 0} = R^n_{\geq 0} \cap H \cap K := D. \tag{3}$$

Deutsch discusses that the computation of a projection onto a collection that consists of limited number of closed and convex sets, performing alternating projection onto the members of that intersection, would be enough (Deutsch 2001).

To perform the projection, consider $L := H \cap K$ which stands for the intersection of $L2$ norm target hyper plane and $L2$ norm hyper sphere. It should be identified for an index set $I \subseteq 1, 2, \ldots, n$ the set $L_I = \{a \in L | a_i = 0 \; \mathit{for\ all}\ i \notin L\}$ implies a subset of points in L that does not include the coordinates with index not in I. This is the method for operating $proj_L$ and $proj_H$ in Algorithm 1 shown in Figure 1. Using the previous description and Hoyer's sparseness measure constraint σ, computing the sparseness enforcing projection is shown in Figure 1.

In this algorithm, the aim was to assess that the projection onto D can be implemented using alternating and cyclic projection onto the predefined structures.

About the projections onto the simplex C, if $x \epsilon R^n$, then a separator $\hat{t} \in R$ exists such that $p := proj_C(x) = max\left(x - \hat{t}.e, 0\right)$ and the element-wise method (Chen and Ye 2011) is used to calculate maximum. In this paper, we always consider $\hat{t} \geq 0$. This hypothesis denotes omitting all inputs in x that are less than $\hat{t}$ after performing projection.



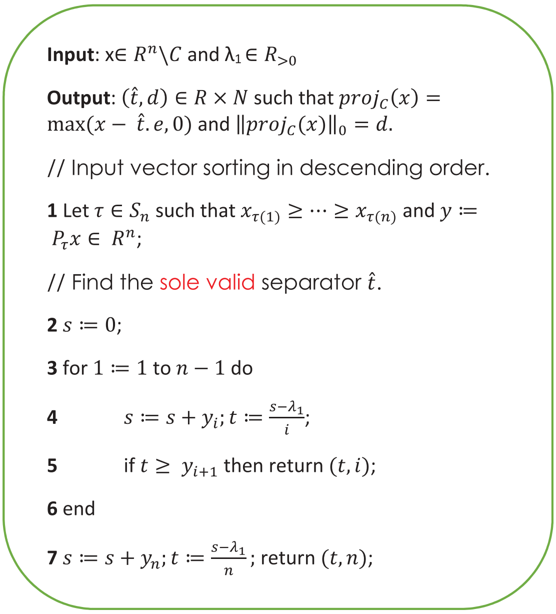
```
Input x ∈ $R^n$ and λ₁ , λ₂ ∈ $R_{>0}$ with λ₂ ≤ λ₁ ≤ $\sqrt{n}$ λ₂
Output s ∈ $proj_D$(x) where D = $S_{\geq 0}^{(\lambda 1 - \lambda 2)}$
// Project onto target hyper plane H and target hyper circle H.
r: = $proj_H(x)$;
s∈ $proj_L$(r);
// repeat projection to find proper solution.
While s∉ $R^n_{\geq 0}$ do
    // Project on to scaled canonical simplex C.
    r ≔ $proj_C(s)$;
    //Project onto L with respect that vanished coordinates at zero.
    s∈ $proj_L$(r) where I: = $\{i \in \{1, \ldots, n\} | r_i \neq 0\}$;
end
```

Figure 1. Algorithm for computing the operator of sparseness enforcing projection in respect to Hoyer's sparseness measure σ that was developed by Thom and Palm (2013).

Figure 2 shows the algorithm used in the present paper to compute the separator $\hat{t}$ and the number of nonzero members while projection onto C that was developed by Thom and Palm (2013). The algorithm presents an adapted version of Chen and Ye's method (Chen and Xiaojing 2011). During the algorithm S_n refers to the symmetric group and P_t presents the permutation matrix correlated with a permutation of $t \in S_n$. The algorithm's procedure can be explained as: sorting the argument x and calculating the mean value of the largest members in x minus the target $L1$ norm λ_1, named $\hat{t}$. According to Blumensath and Davies, the number of relevant inputs for computation of $\hat{t}$ is equal to the $L0$ pseudo-norm of the projection and is discovered by testing every possible value, starting with the largest one and continue descending (Blumensath and Davies 2009). Computational complexity of the presented algorithm in Figure 2 is influenced by sorting the input vector and thus is quasilinear (Thom and Palm 2013).

Sequential forward feature selection

The main point in feature selection is to catch a group of candidate features and then pick out the ones that have the best performance in classification systems. The introduced process is capable of reducing and refining not only the complexity of problems, by reducing the mass computation and collected



```
Input: x∈ $R^n \setminus C$ and λ₁∈ $R_{>0}$
Output: $(\hat{t}, d) \in R \times N$ such that $proj_C(x) = \max(x - \hat{t}.e, 0)$ and $\|proj_C(x)\|_0 = d$.
// Input vector sorting in descending order.
Let $\tau \in S_n$ such that $x_{\tau(1)} \geq \cdots \geq x_{\tau(n)}$ and $y := P_\tau x \in R^n$;
// Find the sole valid separator $\hat{t}$.
$s := 0$;
for $1 := 1$ to $n - 1$ do
    $s := s + y_i; t := \frac{s-\lambda_1}{i}$;
    if $t \geq y_{i+1}$ then return $(t, i)$;
end
$s := s + y_n; t := \frac{s-\lambda_1}{n}$; return $(t, n)$;
```

Figure 2. Algorithm for information computing for performing projections onto C (Thom and Palm 2013).

features, but also there are some situations that cause to obtain better accuracy in classification because of finite sample size effects (Jain et al. 1982). The procedure of selecting a suitable algorithm really depends on the problem, the size and desired recognition rate and computational performance (Ferri et al. 1994).

Formally in feature selection, given a feature set $X = \{xi|i = 1 \ldots N\}$, find a subset Y_M (Equation (4)), with $M < N$, that maximizes an objective function $J(Y)$, ideally $P(correct)$.

$$Y_M = \{x_{i1}, x_{i2}, \ldots, x_{iM}\} = arg \underset{M, i_M}{max} J\{x_i|1 = 1, \ldots, N\}. \tag{4}$$

It is claimed that the sequential forward selection (SFS) can be counted as one of the most simple and fast procedures of feature selection in different areas (Zongker and Jain 1996).

The mechanism of sequential forward feature selection can be summarized as:

Prior to all, SFS earmarks an empty set and sequentially extends features from feature space. The explained action continues to reach a desired (user-specified) subset size. Each iteration progression includes adding a new feature and evaluating that new feature (not all previous added features to the subset). After that, the evaluation operation would be done using the pretended principle function that assesses the feature which persuades the maximum performance improvement of the feature subset if it is included (Marcano-Cedeño et al. 2010).

To perform feature selection in the proposed method we used sequential forward feature selection in the hidden layer of our network to select appreciate features among all features produced in this layer with sparse connection and sparse activity in this layer.

Kullback–Leibler divergence

The Kullback–Leibler divergence (KL divergence) is a non-symmetric measure of the difference between two probability distributions P and Q which is denoted $D_{KL}(P \vee Q)$. In other words, $D_{KL}(P \vee Q)$ is the value of information lost when Q is used to approximate P (Eguchi and Copas 2006). As the KL divergence is a known method, we refrain from further discussion.

Architecture of the proposed method's network

In this section, the implemented network that is a refined and improved version of Thom et al.'s method (Thom and Palm 2013) and consists of an auto encoder in addition to a two-layer neural network that uses sparseness enforcing projection as transfer function and a unit to pick out the best features is described as seen in Figure 3.

In the proposed model, a module was improvised for reconstruction. In this module the input is converted to an internal representation that is built using W_h and this representation is used to reconstruct the input $\tilde{x} \in R^n$ by means of the auto encoder network, so a weight matrix $W \in R^{d*n}$ and a transfer function which is Hoyer's sparseness enforcing projection were obtained. This transfer function guarantees that our internal representation is close to the input sample and neurons are sparsely active and sparsely connected to each other in the network.

This module is called supervised online auto encoder (SOAE) (Thom and Palm 2013). In the proposed network, the aim is $(W_{e_i}) = \sigma_w$, where σ_w is the target degree of sparseness connectivity and the sparse connectivity holds by enforcing W to be sparsely presented

Here $\sigma_w \in (0, 1)$ and W_{e_i} is ith column of W. This condition has been adopted from Hoyer (2004).

After feature extraction level, to find the suitable number of neurons (suitable number of features) that builds the best internal representation of the input, SFS method is applied. In the sequential forward feature selection procedure, first of all, the set of extracted features computed by the network's hidden layer is given to this unit. It is showed that the structure of SFS might be

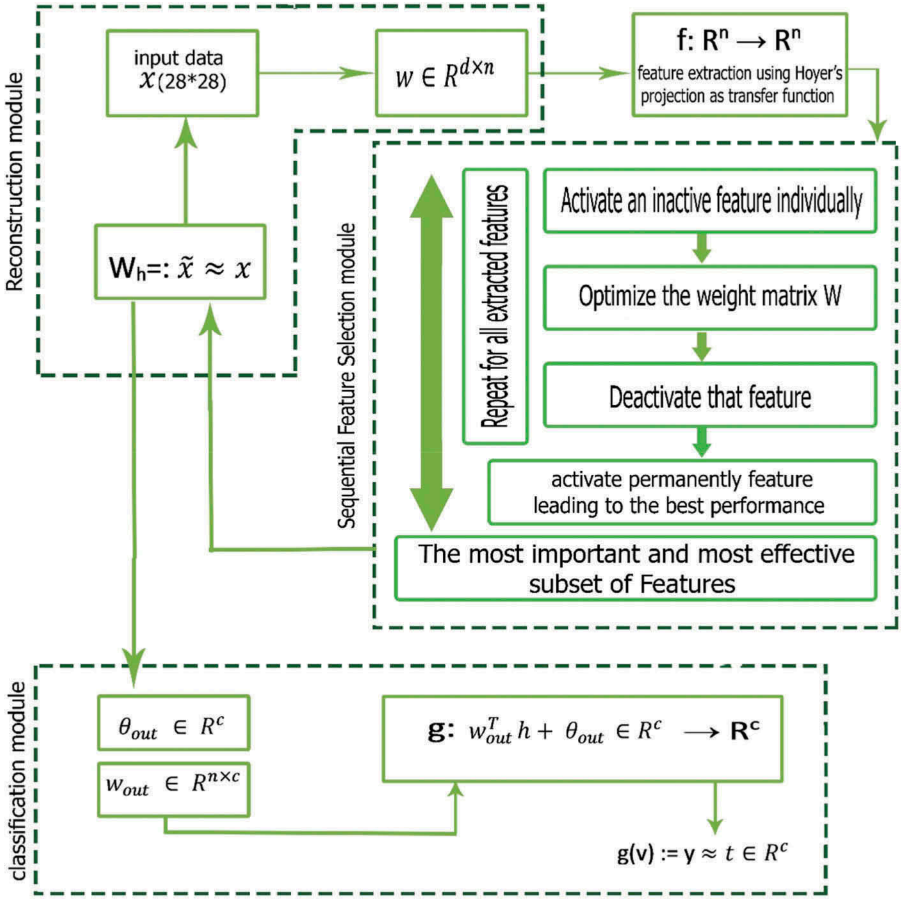

Reconstruction module
input data
$x(28*28)$
$w \in R^{d\times n}$
f: R^n → R^n
feature extraction using Hoyer's projection as transfer function
$W_h =: \tilde{x} \approx x$
Sequential Feature Selection module
Repeat for all extracted features
Activate an inactive feature individually
Optimize the weight matrix W
Deactivate that feature
activate permanently feature leading to the best performance
The most important and most effective subset of Features
classification module
$\theta_{out} \in R^c$
$w_{out} \in R^{n\times c}$
$g: w_{out}^T h + \theta_{out} \in R^c \longrightarrow R^c$
$g(v) := y \approx t \in R^c$

Figure 3. The proposed method's architecture.

divided into three main subgroups that are performing repetitively. It starts with an empty set of features, activates features individually and optimizes the weight matrix concerning the current subset of features. It then deactivates the activated feature to examine its role in the efficiency. During this process the method activates the feature permanently leading to the best performance. The investigated steps are performed for all extracted features and obtain the most important and most effective subset of features which is used to represent data and optimizes weight matrix to achieve high accuracy in classification.

It is obvious that in every iteration, some neurons randomly connect to each other and make a feature. This feature is selected if it improves our representation or omitted if does not have a good role in performance. As a result, the number of neurons in the hidden layer is calculated using sequential forward feature selection algorithm and yields the best effective subset of features which are not redundant and destructive.

The classification module computes the decision by feeding the internal representation (h) through a one layer network. The output of classifier depends on W, W_{out}, θ_{out}. Where W is the synaptic weight matrix of reconstruction part and W_{out} and θ_{out} are the parameters of the classification module and h is just depend on W.

To optimize the parameters explained above, a differentiable similarity measure is used for both reconstruction and classification parts. So, to minimize the deviation in both approximations we have the objective function as Equation (5):

$$E_{SOAE}(W, W_{out}, \theta_{out}) = (1-\alpha).S_R(\tilde{x}, x) + \alpha.S_C(y, t), \tag{5}$$

where y is the approximation of target output (t) and S_R and S_C are similarity measure functions. $\alpha\epsilon[0,1]$ is a tradeoff parameter between classification and reconstruction. If $\alpha = 0$ then the SOAE is a network for finding a good representation to approximating the input and if $\alpha = 1$ then it will just pay attention to the classification part.

This tradeoff variable is adjusted according to $\alpha(v) = 1 - exp(-v/100)$ where $v\epsilon N$ represents the number of the current epoch. Thus it is vivid that the parameter α starts with zero value and gently increases and reaches one. So at the beginning the emphasis is on optimization of reconstruction module and this emphasis decreases slowly on this module and increases for the classification part.

There are some options for similarity functions and here the KL divergence similarity measure is used.

To optimize the proposed objective function, projected gradient descent (Bertsekas 1999) is applied, the parameters are tuned using online learning procedure.

As mentioned before, Hoyer's projection operator is used as hidden layer's transfer function in reconstruction module and sigmoid function for the classification part.

Equivalent to the initialization of the radial basis function networks, presented weight matrix would be attained by choosing a subset of the learning set randomly (Bishop 1995).

Experimental results

To train and test the proposed method, the MNIST data base of handwritten digits was employed (LeCun et al. 1998). This dataset consists of 70,000 samples, including 60,000 learning samples and 10,000 samples for evaluation. In order to generate the original dataset, the displacement of the digits was reached based on their barycenter.

The input data dimension is 28*28 = 784 since each sample in the dataset denotes a digit in a 28*28 image and a class label $c \in 0, 1, \ldots, 9$.

Simard, Steinkraus, and Platt (2003) reported that the employment of 800 hidden units for this dataset yields desired performance when using sufficient learning samples. In the proposed method, the number of neurons in the hidden layer was chosen to be 1000 experimentally and with respect to the introduced information presented above. An initial step size must be set because of the wielding gradient descent algorithm for optimization. Each candidate's step size was ranked by means of performing twofold cross validation five times on the learning data. For each candidate the median of 10 resulting classification errors was then computed and analyzed. Finally, respecting the minimum of the median classification errors, the winning step size was determined. The calculated step size was dampened at every epoch using a factor of 0.999. When the comparative change in the target function was very small and no remarkable refinement was observed in the training set, termination of the optimization process was accomplished. In the next step, the eventuated classifiers were exerted to the evaluation set and the number of wrong classifications was computed. During the experiments it was discovered that 96% of all samples located in the learning set held a sparseness which was less than 0.75 and it lead to set the objective degree of sparse connectivity $\sigma_w = 0.75$, which is why the resulting bases are required to be indeed sparsely connected in comparison with the sparseness of the digits (Thom and Palm 2013). Target degrees of sparse activity σ_H with respect to σ, that is the Hoyer's sparseness measure, are picked from the interval [0.20, 0.95] in steps of size 0.05.

After the training phase, individual samples of the learning set were embedded to the network and active neurons in the hidden layer were enumerated. Furthermore, the achieved sparseness activity was computed for each value of σ_H concerning $L0$ pseudo-norm. The received consequence of mean value and standard deviation of sparse activity is presented in Figure 4. From the figure, it is apparent that there was a net decrease in the standard deviation of activity while the sparseness increased, therefore the mapping from σ_H to the resulting number of active units gets smarter.

For better perception of the method's function, some learned weights of the network are represented in Figure 5.

Table 1 shows the achieved accuracy results of the proposed method in comparison to the achieved accuracy by other methods: SOAE-σ, SOAE-L0, SMLP-SCFC, MLP-OBD and MLP-samples presented in Thom and Palm (2013) and Thom, Schweiger, and Palm (2011).

In Table 1, the last column presents the median ± standard deviation of the attained classification. It can be inferred from the results that the proposed method could best classify the test data in comparison with other similar methods based on the accuracy of classification and evaluation error. In compare with the best results reported in Thom and Palm (2013), the proposed method increase 1% the accuracy of the classification and decreased about 12% the evaluation error. These results show that sequential feature selection can improve the quality of classification.

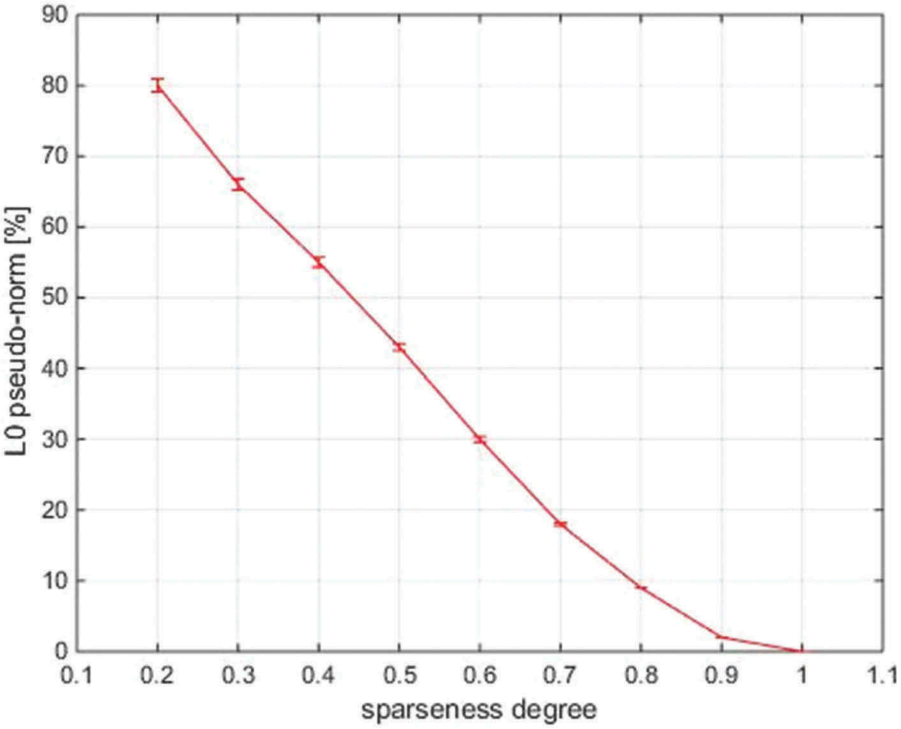


Figure 4. Eventuated number of nonzero records in an internal representation *h* with 1000 entries, with respect to the objective degree of sparseness.

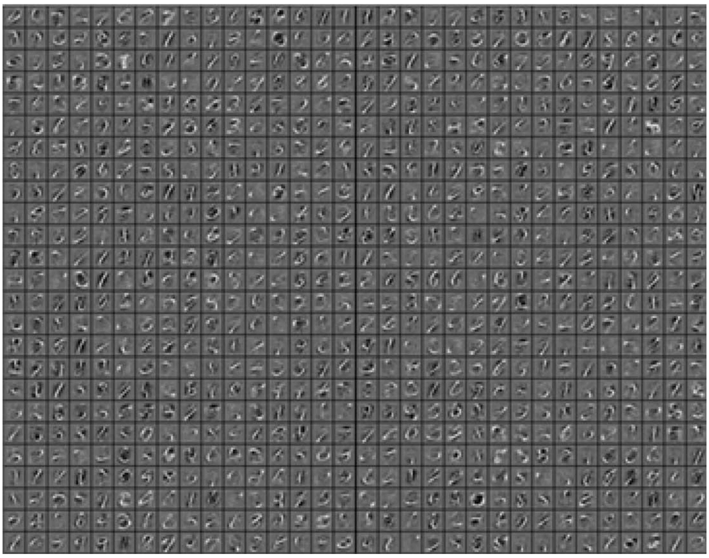


Figure 5. Some learned weights of the network.

Table 1. The achieved results in comparison with some other methods.

Approach	Sparse connectivity	Sparse activity	Accuracy [%]	Evaluation Error [%]
Proposed Method	0.75	[0.45, 0.85]	99.2	63 ± 4
SOAE-σ	0.75	[0.45, 0.85]	98.0	75 ± 4
SOAE-L0	0.75	$k \in [242,558]$	97.7	82 ± 4
SMLP-SCFC	0.75	None	97.7	81 ± 5
MLP-OBD	$\gamma = 12.5\%$	None	98.1	89 ± 4
MLP-samples	None	None	98.0	91 ± 5
MLP-SCFC	None	None	97.9	91 ± 6

Conclusion

The engineers of artificial information processing systems obtained noticeable practical benefits using the sparseness implication which was ascertained by neuroscientists.

In this study, we attempted to build a strong classifier to classify handwritten digits. To achieve this goal, first of all, a study and computation on σ was performed and the Hoyer's sparseness measure and particularly the projection of arbitrary vectors onto sets by means of the achieved value for σ done. The transfer function of the neural network implemented using the σ projection and the characteristics of σ led to yielding a differentiable closed-form expression for presumption of sparse code words which is discussed, in detail, in the manuscript. Besides the explained sparse activity, it was also forced in the proposed network to perform sparse connectivity by means of the σ projection after the presentation of learning samples and it is obvious that because of the smoothness of the projection gradient based methods are befit to be employed for optimization. The SOAE was applied for the pattern recognition part of the method while described the hidden unit of network implemented by means of sparseness enforcing projection as transfer function in the neurons. Afterwards, sequential forward feature selection was applied to select the best features among the created ones using sparse connectivity and sparse activity between neurons and finally the internal representation was extracted with the SOAE passed to a one layer network to classify data. The aforementioned results proved that the proposed method performs suitably.

References

Auffarth, B., M. López, and J. Cerquides. 2010. Comparison of redundancy and relevance measures for feature selection in tissue classification of CT images. In *Advances in data mining. Applications and theoretical aspects. ICDM 2010. Lecture Notes in Computer Science*, P. Perner. eds, 6171. Berlin, Heidelberg: Springer.

Bertsekas, D. P. 1999. *Nonlinear programming*. Belmont, Massachusetts: Athena Scientific Belmont.

Bishop, C. M. 1995. *Neural networks for pattern recognition*. New York: Oxford University Press. 1995.

Blumensath, T., and M. E. Davies. 2009. A simple, efficient and near optimal algorithm for compressed sensing. Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, p.3357–60, Taipei, Taiwan, April 19-24, 2009

Chen, Y., and Y. Xiaojing. 2011. "Projection onto a simplex." *arXiv preprint arXiv:1101.6081*.

Ciresan, D. C., U. Meier, L. M. Gambardella, and J. Schmidhuber. 2010. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation* 22(12):3207–20. doi:10.1162/NECO_a_00052.

Deutsch, F. 2001. *Best Approximation in Inner Product Spaces, volume 7 of CMS Books in Mathematics*. New York: Springer Verlag. 2001.

Donoho, D. L. 2006. For most large underdetermined systems of linear equations the minimal. *Communications on Pure and Applied Mathematics* 59(6):797–829. doi:10.1002/cpa.20132.

Eguchi, S., and J. Copas. 2006. Interpreting Kullback–Leibler divergence with the Neyman–Pearson lemma. *Journal of Multivariate Analysis* 97(9):2034–40. doi:10.1016/j.jmva.2006.03.007.

Ferri, F., P. Pudil, M. Hatef, and J. Kittler. 1994. Comparative study of techniques for large-scale feature selection. *Pattern Recognition in Practice* IV:403–13.

Gregor, K., and Y. LeCun. 2010. Learning fast approximations of sparse coding. Proceedings of the 27th International Conference on Machine Learning (ICML-10), Israel, 2010.

Hoyer, P. O. 2004. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research* 5:1457–69.

Hubel, D. H., and T. N. Wiesel. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology* 160(1):106. doi:10.1113/jphysiol.1962.sp006837.

Hurley, N., and S. Rickard. 2009. Comparing measures of sparsity. *Information Theory, IEEE Transactions On* 55(10):4723–41. doi:10.1109/TIT.2009.2027527.

Jain, A. K., and B. Chandrasekaran. 1982. Dimensionality and sample size considerations in pattern recognition practice. In *Handbook of Statistics*, eds. P. R. Krishnaiah, and L. N. Kanal, Vol. 2, 835–55. Amsterdam: North-Holland. 1982.

Laub, A. J. 2005. *Matrix Analysis for Scientists and Engineers*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM). 2005.

Laughlin, S. B., and T. J. Sejnowski. 2003. Communication in neuronal networks. *Science* 301 (5641):1870–74. doi:10.1126/science.1089662.

LeCun, Y., C. Cortes, and C. Burges, 1998. MNIST handwritten digit database 1998, 2016, [online] Accessed http://www.research.att.com/~yann/ocr/mnist.

Marcano-Cedeño, A., J. Quintanilla-Domínguez, M. G. Cortina-Januchs, and D. Andina. 2010. Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society, Glendale, Arizona, USA.

Markram, H., J. Lübke, M. Frotscher, A. Roth, and B. Sakmann. 1997. Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. *The Journal of Physiology* 500(2):409–40.

Mason, A., A. Nicoll, and K. Stratford. 1991. Synaptic transmission between individual pyramidal neurons of the rat visual cortex in vitro. *The Journal of Neuroscience* 11(1):72–84.

Natarajan, B. K. 1995. Sparse approximate solutions to linear systems. *SIAM Journal on Computing* 24(2):227–34. doi:10.1137/S0097539792240406.

Olshausen, B. A., and D. J. Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research* 37(23):3311–25.

Olshausen, B. A., and D. J. Field. 2004. Sparse coding of sensory inputs. *Current Opinion in Neurobiology* 14(4):481–87. doi:10.1016/j.conb.2004.07.007.

Rehn, M., and F. T. Sommer. 2007. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of Computational Neuroscience* 22(2):135–46. doi:10.1007/s10827-006-0003-9.

Simard, P. Y., D. Steinkraus, and J. C. Platt. 2003. Best practices for convolutional neural networks applied to visual document analysis. 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA.

Theis, F. J., K. Stadlthanner, and T. Tanaka. 2005. “First results on uniqueness of sparse non-negative matrix factorization.” Proceedings of the 13th European Signal Processing Conference (EUSIPCO’05), Antalya, Turkey.

Thom, M., R. Schweiger, and G. Palm. 2011. Training of sparsely connected MLPs. In *Pattern Recognition. DAGM 2011. Lecture notes in computer science*, R. Mester, and M. Felsberg, eds, Vol. 6835. Berlin, Heidelberg: Springer.

Thom, M., and G. Palm. 2013. Sparse activity and sparse connectivity in supervised learning. *Journal of Machine Learning Research* 14(1):1091–143.

Vinje, W. E., and J. L. Gallant. 2000. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science* 287(5456):1273–76.

Zongker, D., and A. Jain. 1996. “Algorithms for feature selection: an evaluation.” Pattern recognition, 1996., Proceedings of the 13th International Conference, Vienna, Austria.