IJICTR
International Journal of Information &
Communication Technology Research

Volume 11- Number 2 – Spring 2019 (29 -37)

Sparse, Robust and Discriminative Representation by Supervised Regularized Auto-Encoder

Nima Farajian
Department of Computer Engineering,
Faculty of Computer and Electrical Engineering
University of Kashan
Kashan, Iran
nimaff2000@yahoo.com

Peyman Adibi[*]
Artificial Intelligence Department, Computer Engineering Faculty
University of Isfahan
Isfahan, Iran
adibi@eng.ui.ac.ir

Received: 15 August 2018 - Accepted: 10 March 2019

Abstract— **Recent researches have determined that regularized auto-encoders can provide a good representation of data which improves the performance of data classification. These type of auto-encoders provides a representation of data that has some degree of sparsity and is robust against variation of data to extract useful information and reveal the underlying structure of data. The present study aimed to propose a novel approach to generate sparse, robust, and discriminative features through supervised regularized auto-encoders, in which unlike most existing auto-encoders, the data labels are used during feature extraction to improve discrimination of the representation and also, the sparsity ratio of the representation is completely adaptive with data distribution. Results reveal that this method has better performance in comparison to other regularized auto-encoders regarding data classification.**

Keywords-component; Supervised Auto-encoder, Feature Learning, Discriminative Representation, Manifold

I. INTRODUCTION

Performance of a machine learning algorithm strongly depends on the features extracted from the data. Traditional feature engineering methods perform well on low-dimensional data. But with increasing advancement of computer sciences in several parts of lifestyles and industry, the data used in computer systems have been expanded significantly related to volume and dimension. These huge data generally have high dimensions and in the preliminary structure they have the least information for discriminating and classifying data. Therefore, traditional feature engineering methods which usually rely on humanistic knowledge for feature extraction, are unable to extract meaningful and structural features from these high-dimensional data [1]. These caused scientists to focus on methods that can produce good features by examining data automatically and without any initial knowledge to improve class discrimination and revealing the underlying structure of data very well. In the machine learning literature, the representation learning refers to these methods and try to learn features which can provide useful and structural information from the data for classification and prediction [2].

Owing to a brain-like hierarchical learning system, deep learning has been the main stream of feature extraction over the past few years. Autoencoder is commonly used as one of the most effective methods of unsupervised feature learning to achieve a deep

[*] Corresponding author.

character hierarchy [3]. In initial idea, auto-encoders have been used to reduce dimensions of data and show them in lower dimensional subspace [4]. However, it was determined in recent research that by making auto-encoder as over-complete, i.e. the number of neurons of hidden layer is more than input layer, the quality of extracted features are improved significantly by adding some regularization terms to the objective function. In the representation learning literature, these auto-encoders are called the regularized auto-encoders [5].

Generally, autoencoders do not use the class information to learn features, which is why they are classified into the category of unsupervised feature learning approaches. However, in new researches, some models of auto-encoders are proposed in which data labels are used during features extraction to improve classification of data. With regard to using data labels, these methods are called supervised auto-encoders [6].

In this research, we propose a new discriminative regularized sparse auto-encoder (DRSAE) which generate sparse, robust and discriminative features and has the following innovations:

- Despite most of popular auto-encoders, data labels are used during feature learning in order to improve classification of data and therefore, the proposed model is placed among supervised auto-encoders.
- The extracted features are robust to small variations around each data and are sensitive to changes along the data manifold.
- The generated features have sparsity characteristic and despite other methods wherein the sparsity ratio should be explicitly determined, in the proposed model the degree of sparsity is adaptive and dynamic with respect to the data complexity and distribution.
- In the presented method, we try to increase the between-class margin while maintaining locality of the within-class data by adding some regularizers, such that the neighboring within-class data are projected near to each other in the feature space while the distance of close between-class data increases and they became apart.

Experimental results on the CIFAR and the MNIST datasets reveal that proposed method has good generalization and better classification than other auto-encoders variants that have been presented so far.

II. Related Works

A. Classic auto-encoders

Auto-encoder is a type of neural networks which are used for unsupervised learning [7] [8]. These networks are composed of three layers of input, encoder and decoder. Data are imported to the network through input layer and a different representation or encoding of input is produced in encoder layer using $f_\theta(x) = S_f(b_e + Wx)$, where S_f, W and b_e denote nonlinear activation function, the weight matrix and bias vector of encoding layer, respectively. Decoder layer acts reversely and decodes information produced in encoder layer to generate $\hat{x} = S_o(b_d + W'h)$. Where S_o, W' and b_d denote nonlinear activation function, the weight matrix and bias vector of decoding layer, respectively. Typically logistic sigmoid ($f(x) = \frac{1}{1+e^{-x}}$) is used for activation function and the weights of encoding and decoding are tied $(W' = W^T)$.

The purpose of training auto-encoders is finding desired parameters $\theta = \{W, b_e, b_d\}$ to minimize reconstruction error between the output of autoencoder and the input which corresponds to minimizing the following objective function:

$$\min_\theta J_{AE} = \min_\theta \sum_k L(x_k, \hat{x}_k) \qquad (1)$$

Here $L(x, y) = \| \hat{x} - x \|^2$ is reconstruction error. The auto-encoder provides a different representation of data in encoder or hidden layer. In basic auto-encoder, typically the number of neurons or features in encode layer is generally less than input layer and thus have been used as a method to reduce the dimensionality of data because of its compact representation of data at hidden layer.

B. Auro-encoder variants

The performance of standard auto-encoder with lower dimensions of encoded features and using linear activity function is very similar to the principal component analysis. Of course, nonlinear activity functions can improve the extracted features; however, the extraction features still suffer from uncovering the underlying data structures that cause good data discrimination. [9]. The lower dimensions of the feature space, is a bottleneck which forces autoencoder to learn meaningful features from input. If this bottleneck is removed and using autoencoder as over-complete (i.e. the number of neurons in the hidden layer is higher than the input), then the autoencoder moves towards learning the identity function. Therefore, researchers focused on adding restrictions on over-complete autoencoder through architectural change or adding a regularizer. These researches are stated under title of regularized auto-encoder [10]. Some of the most important research are presented follow:

One of the first attempt for improving features quality in autoencoder is denoising auto-encoders [11] [12] which uses corrupted version of data as input feeding to the model and try reconstructing original data in decoder layer based on the corrupted one. Corruption is generally an additive Gaussian noise or a binary masking noise and a discrepancy between the output and the original data makes the objective function. This approach encourages model to become robust against noise. In contractive auto-encoder [13] [14] a jacobian regularization ($\|J_{f(x)}\|_F^2 = \sum_{i,j}(\frac{\partial h_j(x)}{\partial x_i})^2$) is added to the objective function which tries to minimize first and second derivative of hidden layer in relation to input. This causes saturation of many hidden layer neurons (i.e. several hidden units are near the extremes of their range, and their derivative is near zero) and makes invariance features against input perturbations. However, combining this regularizer with reconstruction error, counterbalance its effect and give a representation which is sensitive to changes along direction of the data manifold and is invariance in other directions. Another method that has recently been proposed is a sparse auto-encoder that attempts to observe some degree of sparsity in extracted features, resulting in a limited number of active features in the encoder layer. The sparsity of features usually is done in implicit and explicit ways. In implicit method [15] [16] [17], a big part of features is disabled by adding following penalty term to the objective function:

$$J_{sparse} = KL(\hat{\rho}||\rho) = \sum_{i=1}^{d} \rho log \frac{\rho}{\widehat{\rho_\iota}} + (1-\rho) log \frac{1-\rho}{1-\widehat{\rho_\iota}} \qquad (2)$$

where KL(·‖·) denotes Kullback-Leibler divergence between two distributions, d is the number of neurons in hidden layer, ρ is the sparsity parameter which determines the sparsity ratio of features, and $\widehat{\rho_\iota} = \frac{1}{n}\sum_{k=1}^{n} h_i$ is the average of ith hidden unit representation in n training samples. The drawback of this approach is that it does not always provide a sparse representation for all data. In explicit method [18] [19] [20], the auto-encoder disables a certain proportion of neurons during training which leads to same degree of sparsity for all data. Laplacian auto-encoder [21] considers manifold learning approach during of training auto-encoders and generates features by applying a Laplacian graph-adapted regularizer to the objective function to preserve locality in the feature space as following formula:

$$\Omega(f) = \sum_{x \in S} E(x, g(f(x)) + \frac{\lambda}{2} \sum_{x_i, x_j \in S} W(i,j) \|f(x_i) - f(x_j)\|_2^2 \qquad (3)$$

Where f and g are the encoder and decoder function respectively, E is reconstruction error, $\lambda > 0$ balances effect of the regularizer and x_i, x_j are neighboring sample from input data S. HSAE [22] adds hessian and sparsity regularizers to the objective function to produce sparse and robust features which leads to revealing underlying structure of data while maintaining locality in the feature space. LDSAE [23] stacks two Denoising and Sparse auto-encoders with lossless-constraint denoising regularizer which enhances the anti-noise ability and robustness of representation.

C. Supervised auto-encoders

Although Auto encoders are often used as unsupervised, recent researchers have proposed methods for exploiting data labels during feature learning. Authors in [24] presented a supervised auto-encoder for single sample face recognition wherein they tried to extract certain features relative to each person. In this regard, given a set of k classes training images that include gallery images (called clean data) and probe images (called “corrupted” data) and their corresponding class labels were used as training dataset. In this paper, gallery image along with a corrupted picture are fed into two identical auto-encoders (weights and biases are the same) and auto-encoders try to make close representation for these two images as well as reconstructing gallery image from corrupted one in decoder layer. This idea has been improved in [24] which three deep and stacked layers are used for face recognition and gives better results than most presented methods. Limitation of these two methods is that they only can be used in single sample face recognition and can't be used in other data even the general face recognition problem.

Another approach that has been studied recently by researchers is the discriminative auto-encoders [25] which simultaneously try to minimize within-class and maximize between-class scatter by adding a discriminative regularizers $L(e) = tr(S_w(h)) - tr(S_b(h))$ to the objective function which $S_w \; and \; S_b$ are expressed as follows:

$$S_w(h) = \sum_{i=1}^{c} \sum_{h_{i,j} \epsilon i} (h_{i,j} - \bar{h}_i)(h_{i,j} - \bar{h}_i)^T \qquad (4)$$

$$S_b(h) = \sum_{i=1}^{c} m_i (\overline{h_\iota} - \bar{h})(\overline{h_\iota} - \bar{h})^T \qquad (5)$$

$h_{i,j}$is representation for j^{th}sample from class i and $\overline{h_\iota}, \bar{h}$are denoted as mean vector of h_i and h. This regularizer minimizes the distance between each sample representation with the mean vector representation of its class. The key drawback of this method is that it ignores class distribution and considers the mean vectors of classes for the within-class and the between-class.

LMAE [26] is another type of supervised auto-encoders in which discriminative regularizer try to minimize the distance between each sample pre-activation with other within-class and between-class samples and has the following expressions:

$$J_{large-margin} = \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \eta_{k_1,k_2} \left\|W(x_{k_1} - x_{k_2})\right\|^2 +$$

$$\sigma \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \sum_{k_3=1}^{m} \eta_{k_1,k_2} (1 - \tau_{k_1,k_3}) h(s_{k_1,k_2,k_3}) \qquad (6)$$

Here $\eta_{k_1,k_2} = 1$ indicates that $x_{k2}$is target neighbor of x_{k1} , $\tau_{k_1,k_3} = 1$ determines that x_{k3} has the same label as x_{k1} and $h(s_{k_1,k_2,k_3}) = \max(1 + \|W(x_{k_1} - x_{k_2})\|^2 - \|W(x_{k_1} - x_{k_3})\|^2, 0)$ is a slack variable.

LCCSEAE [27] integrates both sparsity regularier and label consistency constraints into the objective function and maximize the intra-class margin through center loss. In [28], another form of supervised auto encoder was proposed that combines reconstruction error and classification error as a single objective function. The input is constructed by fusing noisy concatenated input and label. The experimental results showed its good performance compared to other existing methods. In another research authors introduced the Discriminatively Latent Regularized Variational Auto-Encoder (DLR-VAE) [29] which applied a discriminative regularization on the latent embedding of a variational auto-encoder and investigate its effects on classification and regression.

III. THE PROPOSED METHOD

In this section, we introduce proposed model. It is supposed that input data $D = \{x^1, x^2, \ldots, x^N\}$are in m-dimensional space $x^i \in \mathcal{R}^m$ and auto-encoder encodes data to a d-dimensional space $h^i \in \mathcal{R}^d$ and $> m$ $f_\theta(x) = S_f(b_e + Wx)$. In decoding layer, auto-encoder decodes representation of hidden layer to primary input $\hat{x} = S_o(b_d + W'h)$. Activity functions in both layers are sigmoid and tied weights are used. In Fig. 1 the architecture of the proposed model is shown.

As illustrated in Fig. 1, the presented auto-encoder composed of one hidden layer and one reconstruction layer with sigmoid activation and tied weight. In the training stage, in addition to the input, two other types of data (near-hits, near-misses) are fed into the model and their representations are retrieved. The similarity between those representations is used in the objective function and updating parameters.

The main difference between the proposed model and the basic auto-encoder is the objective function, which consists of three important parts and is expressed as follows:

$$J_{SSS} = J_{AE} + \lambda J_{LP} + \beta J_{Dis} \qquad (7)$$

Where λ, β and γ are the parameters to balance the different regularizers, respectively.

Here $J_{AE} = \sum_{x \in D} L(x, g(f(x)))$ is reconstruction error which exists in all auto-encoders and aims to reduce the difference between input data and their reconstruction.

The locally-preserving regularizer (J_{LP}) has following formula:

$$J_{LP} = \sum_{i=1}^{s} \parallel KBest(f(x_{nearhit}^{i})) - KBest(f(x)) \parallel^2 \qquad (8)$$

KBest(x) is a function that takes a vector as input and its output is a vector with the same size as input which the K maximum elements retain their values, and other elements set to zero, $f(x_{near-hit}^{i})$ is the representation or encoding for the *i^th^* nearest neighbor of x with the same label which is called near-hit of x. Presence of this term in the objective function causes extracted features in hidden layer to have within-class locality-preserving property, and the neighboring input data remain close within the feature space.

$J_{Dis} = -\sum_{i=1}^{s} \parallel KBest(f(x_{nearmiss}^{i})) - KBest(f(x)) \parallel^2$ is the discriminative regularizer and $f(x_{near-miss}^{i})$ is the representation for i^{th} nearest neighbor of x with different label which is called i^{th} near-miss of x. The aim of this term is the opposite of the second regularizer and makes it possible to increase the margin of the neighboring between-class data within the feature space.

The idea of using KBest function is inspired from the k-sparse auto-encoder [30] [31] and Kate [32]. In those works, the authors add a select K best constraint on encoding layer and use this new representation for reconstruction. They show that this method creates a competition between the neurons to get the right for responding to a subset of input data and as a result, makes each neuron specific to a certain structure of input. In our method, we apply KBest function on the J_{LP} and J_{Dis} rather than the reconstruction error to express important discriminative pattern among only competition-winning neurons.

A. The quality of extracted features

The performance of machine learning is heavily depending on the quality of extracted features. The robustness, sparsity and discrimination are three of the most important criteria for evaluating the quality of features. The robustness examines the sensitivity of features against input variations. Usually, two types of variation can occur in data: 1. variations which are perpendicular to the data manifold and don't change nature of data. 2. Variations along the direction of the data manifold which cause movement from one data to another in the data distribution [2] [35]. Robust features are invariance against the first type and in contrast, are sensitive to variations of the second type to be able to reveal data discrimination [13]. In the proposed method, the locally-preserving regularizer make similar representation for the neighboring within-class data to encourage the robustness of features.

This regularizer projects neighboring within-class inputs to a more compact area in the feature space which leads to the invariance of features around each example in the data distribution. However, a very similar representation for all neighbor data causes the representation to be invariance to all directions around inputs which is not appropriate at all. That is why we put restrictions on the equality of just K best features and the rest of features can express specific information of each data and discriminate it from others.

Discrimination is another important measure for feature evaluation which consider inter-class margin in data distribution. Inter-class overlapping often occurs in various datasets and leads to major problems in discrimination and classification. In the suggested method, the discriminative regularizer has been proposed to maximize between-class margin in data and improve data classification. This regularizer targets superior features in the representation and by minimizing it, the distance between each data with its between-class neighbors is increased in the feature space and leads to better discrimination.

The motivation to maximize between-class-margin was also proposed in LMAE. The key difference, however, is that in LMAE linear transformation of samples is considered for increasing margin while in our method, we use k-best items of non-linear transformation of near-miss samples for margin maximization. Also, in LMAE, the large-margin regularizer uses all between-class samples rather than using just k near-miss samples; as it is done in our method which decreases time complexity.

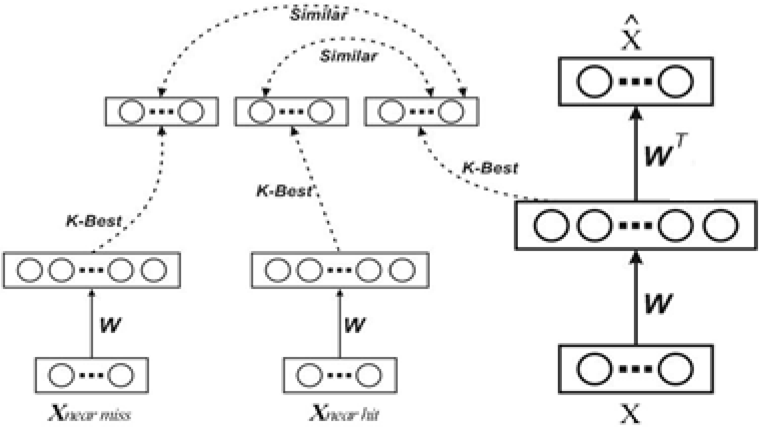


Fig. 1. Architecture of the proposed model (DRSAE) with 1st near hit and 1st near miss

As illustrated in section II, the sparsity of features could improve the quality of features. Sparse Features in auto-encoders means that a limited number of features are active for each data and it forces the auto-encoder to put more specific and important pattern into active features to reconstruct and discriminate data [33] [34]. In the presented model, the sparsity approach is implicit. The presence of J_{LP} and J_{Dis} regularizers encourage the representation to have some degree of sparsity. Given the fact that the locally-preserving regularizer attempts to make k best features of near within-class data become similar, the auto-encoder is forced to disable those features representing the trivial and very specific information of each data and instead, enhance the essential features which are common among the within-class near data.

Also, with regard to the discriminative regularizer which increases distance of the neighboring between-class data, it tries to inactive common features of these data to prevent them located among the k best features. Therefore, the combination of these two regularizers with reconstruction error causes the features which are to some extent common between close within-class data and/or can discriminate the data are maintained while other trivial and very specific features and also common features between neighboring between-class data are disabled. The main difference in this approach with other types of proposed sparse methods is that the sparsity ratio is adaptive and dynamic according to the data distribution.

Finally, the combination of these regularizers with reconstruction error improves quality of features for discrimination and gives sensitive features against variations along the data manifold and makes robustness to other directions. On the other hand, taking into account the sparsity approach of the presented method, those features remain active that reveal changes along the data manifold and are necessary for discrimination and reconstruction, and variations along off-manifold directions are not revealed in active features.

B. *Optimization and Computational complexity*

Auto-encoder performance depends upon its parameters (weight and bias). With respect to training data, the objective function of the neural network is optimized to obtain best values for parameters. In the suggested method, first, all weights are sampled randomly from $U[-b, b]$ where $b = \frac{\sqrt{6}}{\sqrt{N_{in}+N_{out}}}$ which N_{in}, N_{out} are number of neurons in the input and output layer respectively [36]. Bias values are also set equal to zero. In order to maximize the objective function and learn parameters value, stochastic gradient descent optimization is used.

The significant difference in terms of time complexity between the proposed method and the classic auto-encoder is in the k-nn algorithm which is performed on all data once before training to determine near-hit and near-miss of each input. The time complexity of this algorithm is *O(sn2)* where s is the number of near hit and near miss samples and n is number of the input samples. Further, during training, the time complexity of obtaining the regularizers and their gradients w.r.t. the parameters are exactly the same as the reconstruction error which is *O(dmn)* for each iteration. Where *m* is dimensionality of input and *d* is dimensionality of hidden representation [26]. Therefore, the overall time complexity of the presented model is O(sn2+dmn).

IV. EXPERIMENTS AND RESULTS

In this section, two different experiments are performed to precisely assess the proposed model's effectiveness. First experiment examines the quality of feature learning based on criteria described in section III. The second experiment focuses on classification performance of the presented model compared to other popular similar models.

Considered models: based on the literature reviewed in section II, the following autoencoder based models have been chosen for comparison:

- AE: Basic Auto-encoder
- AE + WD: Auto-encoder with weight decay

- DAE-g: Denoising auto-encoder with Gaussian mask noise
- CAE [13]: Contractive auto-encoder
- LAE [21]: Laplacian auto-encoder
- HSAE [22]: Hessian regularized sparse auto-encoder

For all models we used one hidden layer units, tied weight, a sigmoid activation function and the stochastic gradient descent (SGD) was used for optimizing their objective function. All hyper-parameters were tuned on the validation set based on the best classification performance from the candidate set $\{1 \times 10^{e} | \, e = -10, -9, \dots, 10\}$ and the number of near miss and near hit samples was selected among the values (1 to 9) empirically.

In order to evaluate the performance of all models, 5-fold cross-validation was applied, and each experiment was repeated four times to report the average.

Datasets: for all experiments, two benchmark related to image classification are used. The first one is the standard MNIST (hand-written digit classification) [37]. This dataset includes 70000 (28*28) grayscale images of hand-written digits which 50000 are used for training model, 10000 for validation and 10000 for testing. The second dataset is The CIFAR-10 [38] (image classification) which includes 60000 (32*32) RGB images of 10 classes. In our experiments a gray-scale version of the CIFAR (CIFAR-bw) is used which 50000 are used for training, and 10000 for testing.

A. *Feature Learning quality*

In this section, the quality of proposed method's learning features is evaluated. Intuitively the learn features should have more discriminative information and also be robust against some input perturbations. Corruption by noise and affine transformation are from this type of changes and should be ignored in the feature space. Generally visualizing the encoding weights of hidden layer neurons as filter, gives insight information about the quality of features. When learned features are so global, the representation is so sensitive against training set and filters are so similar to the input. In contrast, too local features are so robust to the input perturbations and filters do not factor input into parts. The good representation should be neither too local nor too global to capture underlying manifold of data with enough class discrimination information.

In fig. 2, the visualization of encoding weights of hidden layer is shown for AE, CAE (with high contraction) and the proposed method which are trained on the MNIST. As we can see, the learned features of AE are so global and some filters are the blurred parts of digits (fig. 2a). In CAE the features are too local and all filters are so similar and input are less visible in the filters (fig. 2b). This is due to high contractive degree of CAE which leads to too invariance against input variations. The filters of the proposed method show good balance of locality and globality (fig .2c) These features are appropriate to capture underlying structure of data and contain more information for classification which will be examined in next section.

Another characteristic of good representation is sparsity. As it was mentioned, the presence of certain regularizers in the presented model encourages sparsity of the representation. To evaluate the degree of sparsity of the models, we introduce sparsity ratio measure as follows:

$$Sparsity\ Ratio = \frac{\sum_{i=1}^{N} \frac{N_{Hidden-a}(f(x^i))}{N_{Hidden}}}{N} \times 100 \qquad (9)$$

Where $f(x^i)$ is the hidden layer representation for i^{th} input sample, $N_{Hidden-a}(f(x))$ is the number of active (non-zero) elements in $f(x)$ and N_{Hidden} is the number of neurons in the hidden layer.

Fig. 3 demonstrates sparsity ratio of the proposed model for a various number of hidden units. It can be seen that in the proposed model, the degree of sparsity of features is dynamic and depends on the number of hidden units. This is quite different than many sparse auto-encoders in which the ratio of sparsity is constant.

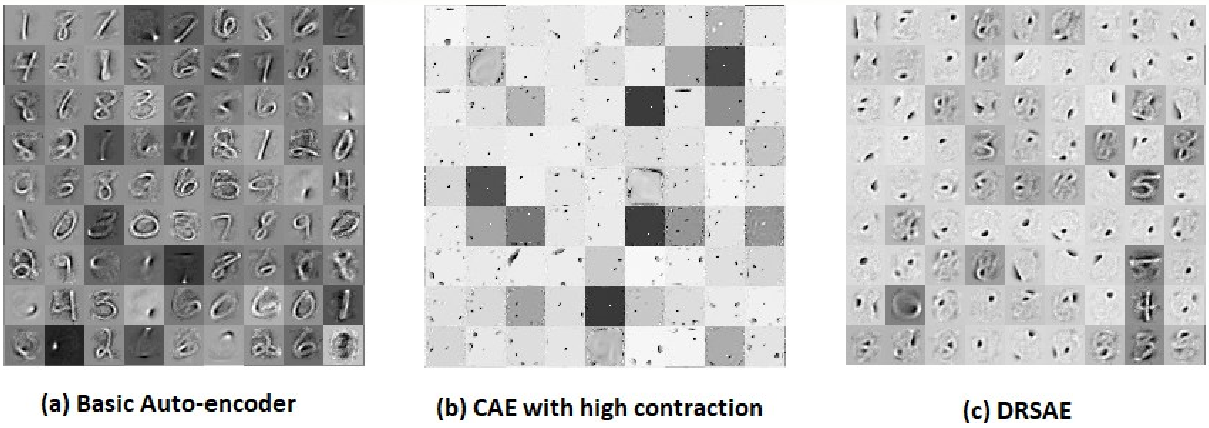
Table1. Sparsity ratio of features of different models on the CIFAR-bw and MNIST

Dataset	Model	Sparsity Ratio
MNIST	Denoising Auto-encoder	49%
	Contractive Auto-encoder	18%
	Proposed Method (DRSAE)	10%
CIFAR	Denoising Auto-encoder	48.5%
	Contractive Auto-encoder	18%
	Proposed Method (DRSAE)	6%

For a better evaluation, the sparsity ratio of our model is compared to CAE and DAE which similar to our model, do not have explicit a sparsity penalty in the objective function as well. In this regard, all models are trained on the CIFAR-bw (1500 hidden units) and the MNIST (1000 hidden units) and the sparsity ratio of extracted features are reported in Table1.

Based on the result, it is clear that the proposed model generates sparser features and despite other models, on the various datasets, the sparsity ratio is different and adaptive with regard to the complexity of data. However, the sparsity of features is useful when the generated features have better performance in term of classification which is considered in the third experiment.

To evaluate robustness of generated features, we compare the average discrepancy of representation between the input image and the changed images by noise and rotation and also the random image and the results are reported in Table2.



(a) Basic Auto-encoder

(b) CAE with high contraction

(c) DRSAE

Fig. 2. Filters of Basic Auto-encoder, CAE with high contraction and DRSAE. These filters are visualized from encoding weights of hidden layer, trained on MNIST.

As it is shown in Table2, the proposed model is robust against some perturbation around each data and also the huge difference between the representations of inputs with the random image, confirms that the generated features well capture the underlying structure of the data distribution.

Table2. The average discrepancy of representation for various perturbation on 100 test samples

Model	25% corruption	50% corruption	10° Rotation	20° Rotation	Random Image
DAE	59.03	92.29	72.03	117.35	234.60
CAE	30.1	78.12	34.03	57.52	143.03
AE	71.11	112.21	136.41	174.16	261.32
DRSAE	**21.46**	**37.24**	**34.06**	**56.48**	**145.48**

B. Classification performance

In this section, the classification performance of the proposed model is compared to other popular auto-encoders variants. Considering that all models are unsupervised, to take advantage of them in classification a pre-train/fine-tune approach are used [39].

In pre-training step, all models are trained with one hidden layer and then in fine-tuning step, a multilayer perceptron network is built and trained in a supervised manner by using parameters (weights and biases) learned from the previous step as initial values and adding a softmax layer on top of the last layer with random weights. Final classification results on the MNIST and the CIFAR-bw are reported in Table 3. As it can be seen, the existence of special regularizers in the objective function of the presented model leads to the representation which makes better discrimination among between-class data and has better classification performance than other models.

Table3. Average Classification performance comparison (Error rate (%)) among different methods on MNIST and CIFAR-bw datasets.

Model	MNIST	CIFAR-bw
AE	1.78	55.47
AE+wd	1.68	55.03
DAE-g	1.18	54.81
CAE	1.14	47.86
HSEA	1.05	46.36
LAE	1.07	46.74
DRSAE	**1.03**	**46.21**

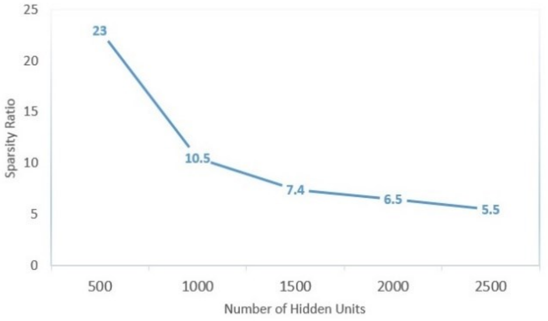


Fig. 3. Sparsity ratio of the representation with different hidden units

In order to compare classifiers and to show whether the performance differences between different classifiers are statistically significant, we have to give the comparison a statistical support [40]. To do so, we use nonparametric tests according to the recommendations made in [41], where a set of proper nonparametric tests for statistical comparisons of classifiers is presented.

Due to the number of datasets and classifiers, the Wilcoxon paired signed-ranks test [42] are performed to find out whether there exist significant differences between a pair of classifiers. This method is widely used for comparing two classifiers on multiple datasets.

Considering 5-fold cross validation on two datasets, 10 different experiments are performed for each classifier. Table 4. shows the results of the Wilcoxon rank-sum test for multiple pairwise comparisons between the purposed method and the other methods with a significance value of $\alpha = 0.05$. As it is shown the null hypothesis is rejected for all comparisons, i.e. the difference between the classifiers does not follow a symmetric distribution around zero.

Table4. WILCOXON Test for Classifier .

DRSAE	Hypothesis ($\alpha = 0.05$)	p-value
AE	Not Rejected	6.402E-9
AE+wd	Not Rejected	3.840E-7
DAE-g	Not Rejected	1.253E-6
CAE	Not Rejected	2.344E-5
HSEA	Not Rejected	4.648E-4

LAE	Not Rejected	1.338E-4

Also HSEA has the best performance among the other methods, but DRSAE outperforms it, and there are significant differences between the two algorithms with a confidence level higher than 95%.

V. Conclusions

In this paper, a new supervised regularized auto-encoder was presented for features generation. Presence of some regularizers in the objective function encourages auto-encoder to generate features which have an appropriate degree of sparsity and are robust against variations around each input. In addition by increasing margin of within-class data, this model enhances discrimination of data. Results show better performance of the proposed model in term of classification in comparison to other similar models.

References

[1] Y. Bengio and others, "Learning deep architectures for AI," Foundations and trends{\textregistered} in Machine Learning, vol. 2, pp. 1-127, 2009.

[2] Y. Bengio, A. Courville and P. Vincent, "Representation learning: A review and new perspectives," IEEE transactions on pattern analysis and machine intelligence, vol. 35, pp. 1798-1828, 2013.

[3] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," in Advances in neural information processing systems, 1994.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," science, vol. 313, pp. 504-507, 2006.

[5] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe and A. Y. Ng, "Measuring invariances in deep networks," in Advances in neural information processing systems, 2009.

[6] R. Huang, C. Liu, G. Li and J. Zhou, "Adaptive Deep Supervised Autoencoder Based Image Reconstruction for Face Recognition," Mathematical Problems in Engineering, vol. 2016, 2016.

[7] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," Biological cybernetics, vol. 59, pp. 291-294, 1988.

[8] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," nature, vol. 323, p. 533, 1986.

[9] N. Japkowicz, S. J. Hanson and M. A. Gluck, "Nonlinear autoassociation is not equivalent to PCA," Neural computation, vol. 12, pp. 531-545, 2000.

[10] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," The Journal of Machine Learning Research, vol. 15, pp. 3563-3593, 2014.

[11] P. Vincent, H. Larochelle, Y. Bengio and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in Proceedings of the 25th international conference on Machine learning, 2008.

[12] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol. 11, pp. 3371-3408, 2010.

[13] S. Rifai, P. Vincent, X. Muller, X. Glorot and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in Proceedings of the 28th international conference on machine learning (ICML-11), 2011.

[14] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin and X. Glorot, "Higher order contractive auto-encoder," Machine Learning and Knowledge Discovery in Databases, pp. 645-660, 2011.

[15] H. Lee, C. Ekanadham and A. Y. Ng, "Sparse deep belief net model for visual area V2," in Advances in neural information processing systems, 2008.

[16] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang and S. Yan, "Sparse Representation for Computer Vision and Pattern Recognition," Proceedings of the {IEEE}, vol. 98, pp. 1031-1044, jun 2010.

[17] S.-Z. Su, Z.-H. Liu, S.-P. Xu, S.-Z. Li and R. Ji, "Sparse auto-encoder based feature learning for human body detection in depth image," Signal Processing, vol. 112, pp. 43-52, 2015.

[18] Y.-l. Boureau, Y. L. Cun and others, "Sparse feature learning for deep belief networks," in Advances in neural information processing systems, 2008.

[19] A. Makhzani and B. Frey, "K-sparse autoencoders," arXiv preprint arXiv:1312.5663, 2013.

[20] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in Advances in Neural Information Processing Systems, 2015.

[21] K. Jia, L. Sun, S. Gao, Z. Song and B. E. Shi, "Laplacian Auto-Encoders: An explicit learning of nonlinear data manifold," Neurocomputing, vol. 160, pp. 250-260, 2015.

[22] W. Liu, T. Ma, D. Tao and J. You, "HSAE: A Hessian regularized sparse auto-encoders," Neurocomputing, vol. 187, pp. 59-65, 2016.

[23] J. Zhang, Y. Zhang, L. Bai, and J. Han, “Lossless-constraint denoising based auto-encoders,” Signal Processing: Image Communication, vol. 63, pp. 92-99, 2018.

[24] S. Gao, Y. Zhang, K. Jia, J. Lu and Y. Zhang, "Single sample face recognition via learning deep supervised autoencoders," IEEE Transactions on Information Forensics and Security, vol. 10, pp. 2108-2118, 2015.

[25] J. Xie, Y. Fang, F. Zhu and E. Wong, "Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval," in Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015.

[26] W. Liu, T. Ma, Q. Xie, D. Tao and J. Cheng, "LMAE: a large margin auto-encoders for classification," Signal Processing, vol. 141, pp. 137-143, 2017.

[27] C. Hu, X.-J. Wu and Z.-Q. Shu, "Discriminative feature learning via sparse autoencoders with label consistency constraints," Neural Processing Letters, pp. 1-13, 2018.

[28] F. Du, J. Zhang, N. Ji, J. Hu and C. Zhang, "Discriminative representation learning with supervised auto-encoder," Neural Processing Letters, vol. 49, pp. 507-520, 2019.

[29] I. Kossyk and Z.C. Márton, “Discriminative regularization of the latent manifold of variational auto-encoders,” Journal of Visual Communication and Image Representation, vol. 61, pp.121-129, 2019.

[30] A. Makhzani and B. Frey, "K-sparse autoencoders," arXiv preprint arXiv:1312.5663, 2013.

[31] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in Advances in Neural Information Processing Systems, 2015.

[32] Y. Chen and M. J. Zaki, "Kate: K-competitive autoencoder for text," in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.

[33] D. Arpit, Y. Zhou, H. Ngo and V. Govindaraju, "Why regularized auto-encoders learn sparse representation?," arXiv preprint arXiv:1505.05561, 2015.

[34] A. Coates, H. Lee and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," Ann Arbor, vol. 1001, p. 2, 2010.

[35] S. Rifai, Y. Dauphin, P. Vincent, Y. Bengio and X. Muller, "The Manifold Tangent Classifier.," in NIPS, 2011.

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010.

[37] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, pp. 2278-2324, 1998.

[38] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[39] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle and others, "Greedy layer-wise training of deep networks," Advances in neural information processing systems, vol. 19, p. 153, 2007.

[40] S. Garcia, A. Fernandez, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability," Soft Computing, vol. 13, no. 10, pp. 959–977, 2009.

[41] J. Demsar, "Statistical comparisons of classifiers over multiple data sets." Journal of Machine learning research, vol. 7, pp. 1–30, 2006.

[42] F. Wilcoxon, "Individual comparisons by ranking methods." In Breakthroughs in statistics, pp. 196-202. Springer, New York, NY, 1992

Nima Farajian received his M. Sc. degree in Artificial Intelligence from Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran in 2012. He is currently a Ph.D. student in Artificial Intelligence in University of Kashan. Additionally, he is a faculty member of computer and electrical engineering department at University of Eyvanekey, where he is the head of Research and Technology. His current research interests include Deep Learning and Neural Network, Machine Learning Pattern Recognition, Computer Vision Image Processing.

Peyman Adibi received the Ph.D. degree from Faculty of Computer Engineering, Amirkabir University of Technology, Tehran, Iran in 2009. He is currently an assistant professor of the Faculty of Computer Engineering at University of Isfahan, where he is the head of the of Artificial Intelligence Department. His current research interests include Machine Learning Pattern Recognition, Computer Vision Image Processing, Computational Intelligence and Soft Computing, Statistical Signal Processing.