Indian Journal of Science and Technology, *Vol 9(27), DOI: 10.17485/ijst/2016/v9i27/97704, June 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Multi-Layer Neural Network Auto Encoders Learning Method, using Regularization for Invariant Image Recognition

Pavel Vyacheslavovich Skribtsov and Pavel Aleksandrovich Kazantsev

PAWLIN Technologies Ltd., Dubna, Moscow Region, Russia; pvs@pawlin.ru, pak@pawlin.ru

Abstract

Background/Objectives: This paper proposes a new type of regularization for deep learning neural networks that is capable of explicit separation of the lower dimensional hidden layer input pattern representation into two components: class information component and transform component. **Methods:** Currently, researchers involved in pattern recognition problems are actively searching for the replacement of deterministic feature extraction algorithms by unsupervised methods capable of generating optimal domain-specific image features during the training process of auto-associative multilayer neural networks. The result of the training process of the deep neural network with a "bottleneck" hidden layer is the task-oriented encoder capable of efficient input signal dimensionality reduction. **Findings:** Many important useful properties of the encoder including the degree of invariance of the feature extraction to input signal transformations (perturbations) greatly depend on the particular form of the regularization applied. In addition to the regular weight decay smoothing component the suggested regularization has two additional components: the first one minimizes the spread of the class-describing features under different pattern transforms and the other component minimizes the spread of the transformation description features for the objects with same perturbations but from the different classes. Class-membership information from the training sequence is used along with the introduced estimator of the similarity of pattern transform to compute the regularization terms. The research reveals that a private case of the suggested regularization corresponds to the well-known Frobenius norm of Jacobian matrix of the *encoder* activations, therefore the contribution of this paper can be seen as a non-local extension of the encoder Jacobian-based family of deep neural network regularizers embedding invariance to non-local input pattern transformations into the deep neural network feature extraction pipeline. Experiments carried out on the synthetic and real pattern datasets show promising results and encourage further investigation of the proposed approach. **Improvements/Applications:** This method can be used for areal images recognition invariant to lighting, weather and orientation, for example for the recognition of vehicles and other landmarks in the images obtained by the unmanned aerial vehicles (UAV).

Keywords: Auto Encoder, Deep Neural Networks, Invariant Image Recognition, Regularization

1. Introduction

Adaptive feature extractors synthesized by unsupervised learning processes provide domain-specialized features for pattern recognition[1] and are considered as a novel alternative to classical deterministic image feature extraction methods, such as Haar-wavelets[2], SIFT[3], SURF[4], LBP[5], HOG[6], MSER[7], etc[8]. This approach has several advantages. First, it avoids empiricism and automatically generates the optimal features adapted for particular object recognition task to be solved. In addition, no new specific feature extraction code has to be written whenever a new pattern recognition problem arises. Second, it avoids deterministic algorithms parameters tuning or searching for the optimal combinations of features that represent different aspects of the object appearances. Application of neural network architectures for the synthesis of feature extractors provides a high degree of computing parallelism[9], which is especially important for embedded systems with increased independence and constraints on com-

**Author for correspondence*

puting and energy resources (robotics, unmanned aerial vehicles). Sequential, layer-by-layer "Deep Learning" process[10] effectively reduces the dimensionality of the input signal and provides a non-local high-level feature extraction mechanism. In order to improve the generalization capability, the optimization problem of auto-associative neural network training (auto encoder[11]) is regularized. The selected regularization influences the degree of the pattern transformation invariance supported by the feature extraction. This paper suggests a new type of regularization that allows the feature extractor to become invariant to the non-local pattern perturbations and receive separate sets of features that describe object class and object transformation.

1.1 Related Work

In addition to the classical regularization methods (weight decay[12], sparsity constraint[13]) more "advanced" approaches have been recently researched. The paper[14] provides the regularization based on the "top down" principle for the deep belief networks, representing a cascade of restricted Boltzmann machines (RBM[15]). RBM regularization by topographical principle is proposed in[16] to provide local invariance to shifts, rotations, and colour change for the recognition of colour images. The paper[17] proposes regularizer for deep neural networks (DNN) based on the kernel methods. The work[18] is of particular interest, because it explores the possibility of applying the "explicit" constraint to ensure feature independence on local pattern perturbations by minimizing the Frobenius norm of the hidden layer representation Jacobian.

$$\left\| J_f(\bar{x}) \right\|_F^2 = \sum_j \left(\frac{\partial h_j(\bar{x})}{\partial x_i} \right)^2 \qquad (1)$$

where $h_j(x)$ is j-*th* value of the nonlinear (hidden) layer representation, performing input image dimensionality reduction;

x_i - the i-*th* component of the input vector.

The idea of the authors was that this regularization term imposes restrictions on the learning process in such a way, that hidden layer representation of the input image is unchanged in case of small variations (distortions) of the input signal in the Euclidean sense. This approach is local, that is, it is not capable to provide the stability of representation with considerable distortions of the input pattern. Practical problems of image recognition, however, often require stable classification of the objects subjected to significant changes of the object's pose, scale, lighting, etc.

Similar to other approaches, regularization term mentioned above (1), provides contracting nonlinear transformation of the input feature space in order to ensure similarity of encoding vectors (representations) in case of a "permissible" small variation of the input signal. In this case, however, it is difficult for the system to distinguish between the input images transformations, which represent the variation within the group of objects belonging to the same class, and those representing the fundamental difference between the objects of different natures. For example, relatively small Euclidean change of the "1" pattern can turn this object class to "7" by changing the direction of the short figure cap, whereas a small shift of the same pattern in the horizontal direction would induce great change of the vector under the Euclidean metric. The obtained descriptions of the objects in the space of reduced dimension, although to a lesser extent (due to the regularization), still contain part of the object transformation description, since this information is necessary for the input signal reconstruction, which in turn is a major optimization goal for all auto-associative learning algorithms. Information about the pattern transformations (distortion) in this approach turns out to be inseparable from the information about the pattern class and is assigned to representation parameters in unknown way. Another problem with regularization (1) is that serious contradiction occurs – on the one hand, the auto-associative learning algorithm requires perfect reconstruction of the input signal, and on the other hand, the hidden representation is forced by condition (1) to be cleared of object transformations information that prevents from perfect reconstruction.

2. Concept Headings

2.1 Separated Hidden Representation

This paper investigates the possibility of explicit separation of the hidden representation (encoding vector) on two principally different parts, that is, one part responsible to the object class features representation (the p-parameters) and object transformation features (q-parameters) applied to the input object (see Figure 1).

Since the degree of nonlinearity of the transformation carried by the autoencoder with one hidden layer may be

insufficient for such a separation, this approach requires the use of recursive "Deep Learning" method for the synthesis of multi-layer structure (see Figure 2).

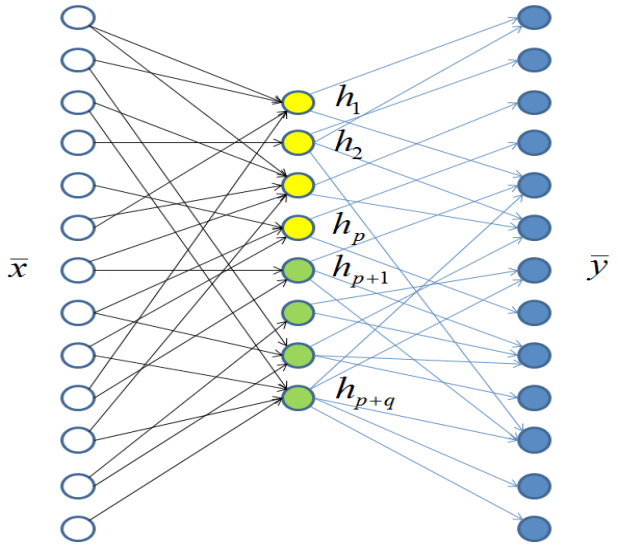

Figure 1. Auto-associative neural network (only part of cross-links are shown) with the encoding layer h, the first p-components (neurons) of which are responsible for the object class representation, and the subsequent q-components are responsible for the information about the transformations applied to the object.

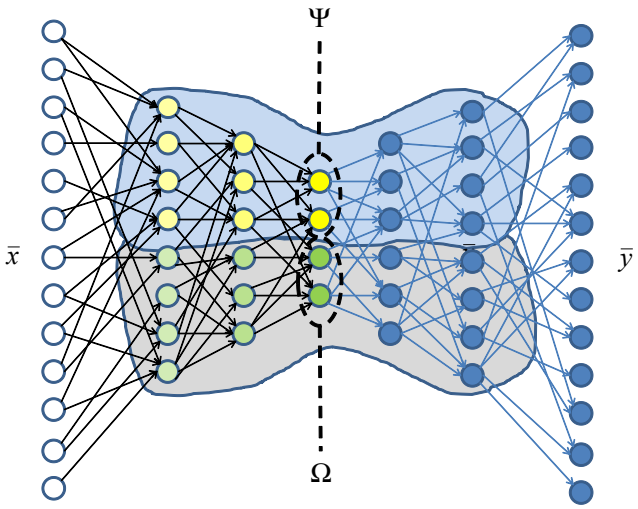

Figure 2. Multilayer auto-associative neural network (only selective cross-links are shown), in which the input image hidden representation separation is improved from layer to layer, ideally reaching the complete separation of information about the class (set of features Ψ) and transformation of the object (set of features Ω) at the bottom (deepest) level of the encoder.

This approach is aimed to reach several goals. First, the existence of such representation would allow a trivial construction of invariant object recognition using only the p-part of the encoded information about the object, containing information just about the object class, not the transformation applied to the image. Second, on each layer the information about transformations can be used to refine the p-parameters on the next layer. Third, the presence of parameters describing the object transformation permits artificial "intervention" (i.e., changing values of the q-parameters) into the reconstruction of the object appearance thus enabling to obtain the image of the object under conditions which were not observed in reality using "analogs" (examples of the similar object transformations in the training sequence).

Auto-associative encoder with one hidden layer can be described by the expression

$$\begin{aligned} \bar{y}(\bar{x},\bar{w}) &= \varphi\left(D\bar{h}\right) \\ \bar{h} &= \varphi\left(H\bar{x}\right) \end{aligned} \quad (2),$$

where (biases are omitted without loss of generalization)

$\bar{x} \in \mathrm{R}^N$ – input signal (image pixels or other high-dimensional features)

$\bar{y} \in \mathrm{R}^N$ – output signal of the network of the same number of dimensions as the input signal,

$\bar{w}$ – weight coefficients of the neural network, including the encoding layer matrix coefficients $H \in \mathrm{R}^{(p+q)\times N}$ and decoding layer matrix $D \in \mathrm{R}^{(p+q)\times N}$, as well as the neuron bias weights, φ is the elementwise nonlinear activation function. In this work a symmetric sigmoid activation function $\varphi(x) = \frac{1}{1+\exp(-x)} - \frac{1}{2}$ was used to minimize the computational complexity.

The intermediate output of the encoding part of the auto-associative network (encoder) can be described by the vector $\bar{h}(\bar{x},H)$. We assume that the first p-components of this vector are responsible for the object class information, and the following q-components for the information about the transformations (perturbations) which could have been applied to the object (change of light, position, etc.).

Autoencoder training can be done by solving the optimization problem with additional regularizing components computed using class membership information and transform similarity estimation (see below):

$$\overline{w}^* = \arg\min_{\overline{w}}\left\{\sum_i \left\|\overline{y}(\overline{x}_i,\overline{w})-\overline{x}_i\right\|^2 + L_1(\lambda,\overline{w}) + L_2(\alpha,\overline{w}) + L_3(\beta,\overline{w})\right\}$$

$$L_1(\lambda,\overline{w}) = \lambda\left\|\overline{w}\right\|^2$$

$$L_2(\alpha,\overline{w}) = \alpha \sum_{\forall i,j:c_i=c_j}\left\|\overline{h}^p(\overline{x}_i,\overline{w})-\overline{h}^p(\overline{x}_j,\overline{w})\right\|^2$$

$$L_3(\beta,\overline{w}) = \beta \sum_{\forall i,j:\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left\|\overline{h}^q(\overline{x}_i,\overline{w})-\overline{h}^q(\overline{x}_j,\overline{w})\right\|^2 \quad (3)$$

where

$\overline{w}^*$ – desired weight coefficients of the neural network,

$\{\overline{x}_i, c_i\}$ – training samples ($\overline{x}_i \in \mathrm{R}^N$ – feature vector, c_i – class number),

θ – positive constant, $\lambda,\alpha,\beta \in \mathrm{R}$ – regularization coefficients.

$\overline{h}^p = \begin{pmatrix} h_1 \\ h_2 \\ \dots \\ h_p \end{pmatrix}$, $\overline{h}^q = \begin{pmatrix} h_{p+1} \\ h_{p+2} \\ \dots \\ h_{p+q} \end{pmatrix}$ - the p-, q-components of the code image representation,

$\sum\left\|\overline{y}(\overline{x}_i,\overline{w})-\overline{x}_i\right\|^2$ – the first error function term, which is the error of auto-association. This "requires" the auto encoder output to coincide with the input. The sum is taken over all training examples.

$L_1(\lambda,\overline{w}) = \lambda\left\|\overline{w}\right\|^2$ – regularization of the representation "simplicity" (in this case using weight decay) may be alternatively replaced by the representation sparsity constraint[13] (usually, the representation component from some small fixed value is calculated using the Kullback-Leibler distance (divergence) from the sum) and, in the latter case, it has the form:

$$L_1(\lambda,\overline{w}) = \sum_{n=1}^{p+q}\rho\log\frac{\rho}{\hat{\rho}_n} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_n},$$

where, in the case of the particular activation function used in this paper

$$\hat{\rho}_n = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{2}+\varphi(\overline{w}_n\overline{x}_i + w_{n0})\right)$$

$L_2(\alpha,\overline{w}) = \alpha \sum_{\forall i,j:c_i=c_j}\left\|\overline{h}^p(\overline{x}_i,\overline{w})-\overline{h}^p(\overline{x}_j,\overline{w})\right\|^2$ – consistency regularization of the p-components of representation in the hidden layer. This component "demands" that the training examples representing objects of the same class with different distortions had similar representation in the p-components of the hidden layer output. There may be too many training pairs of objects with identical classes, so in practice one can take a random subset.

$L_3(\beta,\overline{w}) = \beta \sum_{\forall i,j:\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left\|\overline{h}^q(\overline{x}_i,\overline{w})-\overline{h}^q(\overline{x}_j,\overline{w})\right\|^2$ – consistency regularization of the q-representation, requiring the q-component of the representation to be the same for different training examples for which it can be assumed that objects were subject **to the same parametric distortions** from the "ideal" or "reference" state (same shift, rotation, blur, line thickness change, brightness and contrast, colour palette temperature change, etc.). It is assumed here that objects from different classes may have similarity in the original input space if they were subjected to the same transformations. For example, two identically rotated faces of different persons may have even smaller Euclidean distance in the original (input) feature space than faces of the same person but rotated differently.

$\omega(\overline{x}_i,\overline{x}_j)$ – object transformation similarity measure evaluation function. Small values of the function mean the maximum difference of perturbation; large values mean that the transformations match. The function $\omega(\overline{x}_i,\overline{x}_j)$ must be chosen carefully for specific problems and can use different assessment methods, e.g. use low-frequency spatial component of the image to assess the distribution of light and shifts relatively to the background, use the moments[19], use the oriented gradients histograms[20] to determine the basic orientation for the objects of mostly linear form, use the model-fitting methods or neural network assessments of the object position[21-25]. In some cases, objective information on the object position may be available, taken from sensors during the construction of a data set or during the generation of data sets on synthetic models. It is interesting to consider the special case of $\omega(\overline{x}_i,\overline{x}_j)$ function as a measure for match of the arguments, i.e. $\omega(\overline{x}_i,\overline{x}_j) = \frac{1}{\left\|\overline{x}_i-\overline{x}_j\right\|^2}, \overline{x}_i \neq \overline{x}_j$. At first

glance, such choice may not seem to respect the definition, however, for a large range of applications, the square of the vectors difference norm representing the pixels of the original images will differ less if the objects are subject to the same transformation than if they belong to the same class. For example, the distribution of light on the human face and its position and orientation have a greater effect on the Euclidean images vector difference than the difference in the personalities. In this case, if θ >> 1, then such pairs of (i,j) will be selected that $\left\|\bar{x}_i - \bar{x}_j\right\|^2 < 1$.

Therefore, the denominator will be a small number (i.e. similar or slightly varying training examples will be used). In this case the term $L_3(\beta, \bar{w})$ in the error function corresponds conceptually to regularizer (1), as it will represent the sum of squares of function $\bar{h}^q(\bar{x}, \bar{w})$ component variations divided by the small increment of the argument, which looks like finite difference approximation of the Frobenius Jacobian metric. However there are two major differences. First, in our case this condition will be applied only for q-components. Second, unlike the regularizer (1), ours allow selection of the function $\omega(\bar{x}_i, \bar{x}_j)$ to reflect the non-local transformations (distortions) of the image, which must comply with the same q-parameters sets.

The proposed training process can be called semi-unsupervised since labels for all object classes are not required. In this case, class-similarity regularization component $L_2(\alpha, \bar{w})$ uses only pairs of objects with available class membership information. As the number of object pairs is proportional to the square of the object number excessive computational costs may cause difficulties. This problem can be solved using the following approximation for regularization components (see Figure 3).

Since the independent estimation of the class membership and transformations represents inherently nonlinear problem, it cannot be generally solved by a single-layer neural network. Too large values of common regularization factors α , β can "harm" the optimization process because their high values would actually require to search for non-existing solution. In practice too high values of regularization coefficients lead to poor reconstruction error, which spoils the solution. Therefore, it is important to gradually increase the regularization coefficients, from layer to layer, thus smoothly forming a divided representation of the input image on the p- and q-components.

2.2 The Training Algorithm

Neural network weight updates are made according to the "gradient descent" rule:

$$\bar{w}' = \bar{w} - \eta \frac{\partial}{\partial \bar{w}} \sum_i \left\| \bar{y}(\bar{x}_i, \bar{w}) - \bar{x}_i \right\|^2 - \lambda \frac{\partial}{\partial \bar{w}} L_1(\bar{w})$$

$$- \alpha \frac{\partial}{\partial \bar{w}} L_2(\bar{w}) - \beta \frac{\partial}{\partial \bar{w}} L_3(\bar{w}) \qquad (4)$$

where

$$\eta \frac{\partial}{\partial \bar{w}} \sum_i \left\| \bar{y}(\bar{x}_i, \bar{w}) - \bar{x}_i \right\|^2 = 2\eta \sum_i \left(\bar{y}(\bar{x}_i, \bar{w}) - \bar{x}_i \right) \frac{\partial}{\partial \bar{w}} \bar{y}(\bar{x}_i, \bar{w})$$

– the main gradient component, minimizing the input image reconstruction error using gradient descent, convergence speed parameter η , calculated using the standard or upgraded back propagation procedure, for example, such as RPROP[26].

$\lambda \frac{\partial}{\partial \bar{w}} L_1(\bar{w})$ – the component responsible for the smooth presentation, in the case of using the weight decay regularizer, has a simple form $2\lambda \bar{w}$ and, in the case of using the Kullback-Leibler criterion of representation sparsity for the applied sigmoid activation function of $\varphi(x) = \frac{1}{1+\exp(-x)} - \frac{1}{2}$ neurons have the form of:

$$\frac{1}{M} \sum_{n=1}^{p+q} \sum_{i=1}^{M} \varphi'(\bar{w}_n \bar{x}_i + w_{n0}) x_{n,i} \left(-\frac{\rho}{\hat{\rho}_n} + \frac{1-\rho}{1-\hat{\rho}_n} \right)$$

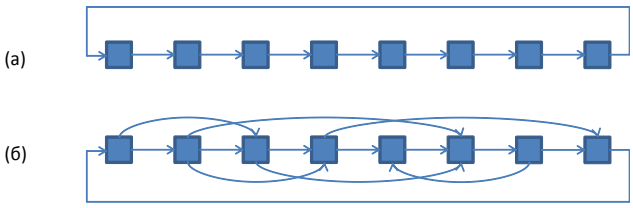

Figure 3. Conceptual view of regularization components approximation, which contain the pairwise differences norms: **(a)** "looping" and **(b)** "looping with screed" **(a)** "looping" is a chain of all objects of the same class with looping is built. In the limit, if the value of the regularizer is equal to zero, the sum of the pairwise differences norms turn to zero as well. In practice, however, this condition for the distant positions in the chain may be poor – with little error between the adjacent elements, the difference may increase along the chain. Therefore an alternative is offered below. **(b)** "looping with screed" - the pairs (neighbours), formed by the "looping" method, are added with a certain amount of random pairs.

Since the derived activation function can be expressed in terms of itself, the regularization component gradient requires calculation of the hidden layers outputs only.

$$\varphi'(x)=\frac{\exp(-x)}{(1+\exp(-x))^2}=\left(\frac{1}{\varphi(x)+\frac{1}{2}}-1\right)\left(\varphi(x)+\frac{1}{2}\right)^2=$$

$$=\left(\frac{\frac{1}{2}-\varphi(x)}{\varphi(x)+\frac{1}{2}}\right)\left(\varphi(x)+\frac{1}{2}\right)^2=\left(\frac{1}{2}-\varphi(x)\right)\left(\varphi(x)+\frac{1}{2}\right)=\frac{1}{4}-\varphi^2(x)$$

$\hat{\rho}_n=\frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{2}+\varphi(\overline{w}_n\overline{x}_i+w_{n0})\right)$ – sparsity measure of the n-*th* component of the hidden representation vector averaged over all elements in the dataset.

$\alpha\frac{\partial}{\partial\overline{w}}L_2(\overline{w})$ – gradient component responsible for the coherent representation of class membership information. The easiest way to represent its structure is to use the form $\alpha\frac{\partial}{\partial w_{t,s}}L_2(\overline{w})$, where the t-index denotes the number of input neuron, and the s-index denotes the number of weight associated with the s-component of the input vector (this gradient is zero over the output layer weights).

$$\frac{\partial}{\partial w_{t,s}}L_2(\overline{w})=\sum_{n=1}^{p}\frac{\partial h_n}{\partial w_{t,s}}=\sum_{n=1}^{p}\frac{\partial}{\partial w_{n,s}}\sum_{i,j:c_i=c_j}\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)=$$

$$=\sum_{n=1}^{p}2\delta_{n,t}\sum_{i,j:c_i=c_j}\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)\left(\varphi'(\overline{w}_n\overline{x}_i)x_i^s-\varphi'(\overline{w}_n\overline{x}_j)x_j^s\right)=$$

$$=2\sum_{i,j:c_i=c_j}\left(\varphi(\overline{w}_t\overline{x}_i)-\varphi(\overline{w}_t\overline{x}_j)\right)\left(\varphi'(\overline{w}_t\overline{x}_i)x_i^s-\varphi'(\overline{w}_t\overline{x}_j)x_j^s\right),$$

for weights-biases

$$\frac{\partial}{\partial w_{t,0}}L_2(\overline{w})=\sum_{n=1}^{p}\frac{\partial h_n}{\partial w_{t,0}}=\sum_{n=1}^{p}\frac{\partial}{\partial w_{t,0}}\sum_{i,j:c_i=c_j}\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)=$$

$$=\sum_{n=1}^{p}2\delta_{n,t}\sum_{i,j:c_i=c_j}\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)\left(\varphi'(\overline{w}_n\overline{x}_i)-\varphi'(\overline{w}_n\overline{x}_j)\right)=$$

$$=2\sum_{i,j:c_i=c_j}\left(\varphi(\overline{w}_t\overline{x}_i)-\varphi(\overline{w}_t\overline{x}_j)\right)\left(\varphi'(\overline{w}_t\overline{x}_i)-\varphi'(\overline{w}_t\overline{x}_j)\right)$$

where $w_{t,s}$ denotes the weight of the hidden layer t-neuron to the s-*th* component of the input vector $w_{t,0}$ (gradient is zero with respect to other weight coefficients).

$\beta\frac{\partial}{\partial\overline{w}}L_3(\overline{w})$ has a similar form, with the only difference being that the weight function $\omega(\overline{x}_n,\overline{x}_m)$ is added and the condition for choosing pairs i, j is changed.

$$\frac{\partial}{\partial w_{t,s}}L_2(\overline{w})=\sum_{n=1}^{p}\frac{\partial h_n}{\partial w_{t,s}}=\sum_{n=1}^{p}\frac{\partial}{\partial w_{n,s}}\sum_{\forall i,j\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)=$$

$$=\sum_{n=1}^{p}2\delta_{n,t}\sum_{\forall i,j\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)\left(\varphi'(\overline{w}_n\overline{x}_i)x_i^s-\varphi'(\overline{w}_n\overline{x}_j)x_j^s\right)=$$

$$=2\sum_{\forall i,j\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left(\varphi(\overline{w}_t\overline{x}_i)-\varphi(\overline{w}_t\overline{x}_j)\right)\left(\varphi'(\overline{w}_t\overline{x}_i)x_i^s-\varphi'(\overline{w}_t\overline{x}_j)x_j^s\right),$$

for weights-biases

$$\frac{\partial}{\partial w_{t,0}}L_2(\overline{w})=\sum_{n=1}^{p}\frac{\partial h_n}{\partial w_{t,0}}=\sum_{n=1}^{p}\frac{\partial}{\partial w_{t,0}}\sum_{\forall i,j\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)=$$

$$=\sum_{n=1}^{p}2\delta_{n,t}\sum_{\forall i,j\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left(\varphi(\overline{w}_n\overline{x}_i)-\varphi(\overline{w}_n\overline{x}_j)\right)\left(\varphi'(\overline{w}_n\overline{x}_i)-\varphi'(\overline{w}_n\overline{x}_j)\right)=$$

$$=2\sum_{\forall i,j\omega(\overline{x}_i,\overline{x}_j)>\theta}\omega(\overline{x}_i,\overline{x}_j)\left(\varphi(\overline{w}_t\overline{x}_i)-\varphi(\overline{w}_t\overline{x}_j)\right)\left(\varphi'(\overline{w}_t\overline{x}_i)-\varphi'(\overline{w}_t\overline{x}_j)\right)$$

3. Results

For the initial proof of the concept a synthetic dataset was generated that represents noised digits from "0" to "7" with distortions in the form of the superposition of shifts along the two axes, blurring and intensity inversion. The size of images was 32x56 pixels. Before submitting images to the neural network input simple zero mean and min-max normalization was carried out that transformed all input vector components to the range [-0.5,+0.5]. Visualization of the reconstructed images was made using the inverse process to convert the output signal to plausible pixel greyscale values [0.255]. Examples of character images of the test and training sequences are shown in Figure 4.

In order to improve the reliability of the results JPEG-artefacts with increased contrast were added to the dataset images.

The autoencoder architecture was chosen to be a regular multilayer perceptron (MLP) with the sigmoid activation function. Selection of operating ranges for scale of the regularization parameters was carried out experimentally by monitoring the true error values of the components of the optimized function during the error minimization by the gradient descent. In case if the regularization term showed no error decrease, its scaling factor was increased. If, however, it prevented the main reconstruction error, then the regularization coefficients were relaxed. The use

of sparsity constraint for the representation based on the Kullback-Leibler measure did not produce any tangible benefits compared with the minimization of the square norm for the vector of weight coefficients, but in terms of computational cost that was much less profitable than simple regularization, therefore in our experiments we used weight decay.

A single-layer autoencoder works very well for the reconstruction of characters subjected to various transformations, including the most of the examples from the test sample, at that making the correct reproduction of input images distortions (shifts, inversion, and thickness change). Figure 5 shows examples of the first auto encoder layer reducing the dimension for the feature space from 1792 to 128.

Weight coefficients of the first layer have the same dimension as that of the input images (offsets excluded) therefore they may be visualized (see Figure 6).

Noise in the weight coefficients of the first layer can be removed by an additional regularization requiring similarity of adjacent weights in terms of the input image topology. However, using this regularizer had no impact on the reconstruction error and minimization of main and auxiliary regularization components, so it was not applied in the final setup.

The synthetic dataset allows generating pairs of indices for the second and third regularization components in the explicit form. In this case, the similarity representation function $\omega(\bar{x}_i, \bar{x}_j)$ is equal to 0 for the pairs of training examples with different transformations and is 1 for pairs of examples transformed in the same way. Training and test set consists of 2,048 examples of objects. Test and training sets differ in the type of the font used and in the line width/contrast. The practical experiments revealed the importance of the first regularization

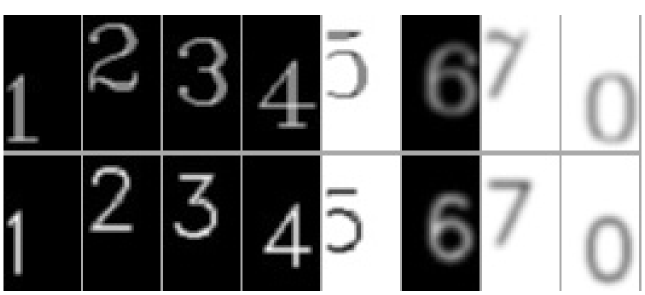

Figure 4. Examples of character images of the test and training sequences before applying noises. Test and train sequences differ in font and contrast.

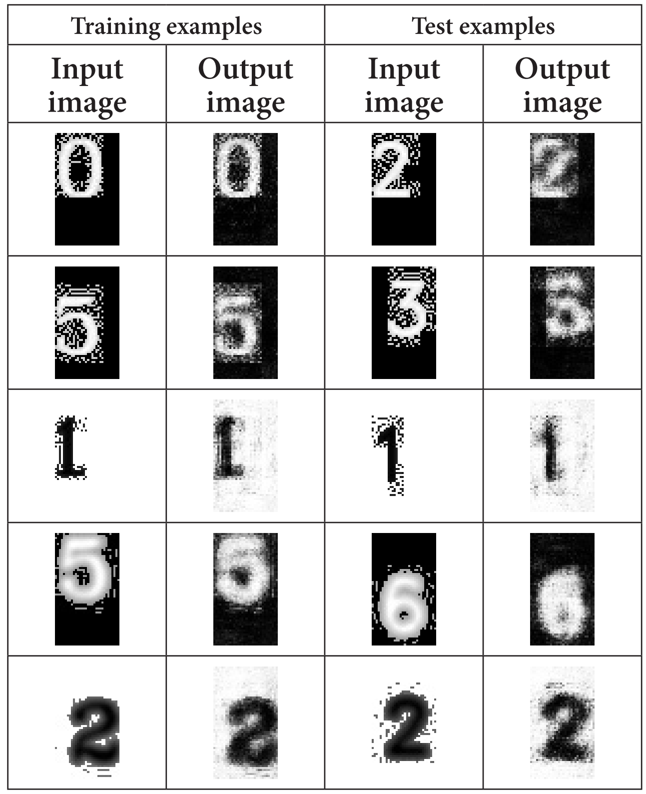

Figure 5. Examples of transformations carried out by the first layer of autoencoder.

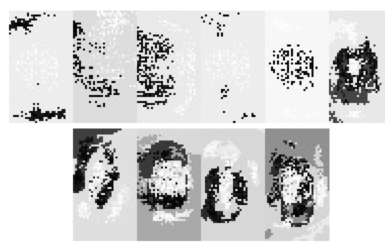

Figure 6. Selective visualization of weight coefficients of the first auto encoder layer.

component ("weight decay" or "sparsity constraint"). In practice, for small parameter values λ, the solutions for the weight coefficients of the autoencoder neural network are also small in the absolute value. At that, the encoder layer forms a conversion close to affine transformation with strong compression and reduction of dimension. The decoder layer, due to large absolute weight coefficients, restores the scale. This minimizes the second ($L_2(\alpha, \overline{w})$) and the third ($L_3(\beta, \overline{w})$) regularization components, without making any useful separation of the p- and q-components. Application of the first regularization

component imposes a restriction on the absolute values of the weight coefficients of the decoding layer, thus "forcing" the hidden layer of the neural network to perform the "mental" work. The scaling coefficients at the second and third regularization components are important as well. Too large values have adverse effect on the reconstruction error, while too small values reduce the class-transform separation effect. During the experiments it was found that the best approach is to gradually increase the values of the main regularization coefficients from a layer to a layer. In this case, each subsequent layer performs more and more work on separation of the representation on class and distortion information. During experiments achievement of separation succeeded in structures with more than 4 layers. Averaged graphs (for a variety of experiments) of changing values of the second and third regularization components from a layer to a layer are shown in Figure 7 for the two cases – at a constant value of a regularization component and at a monotonically

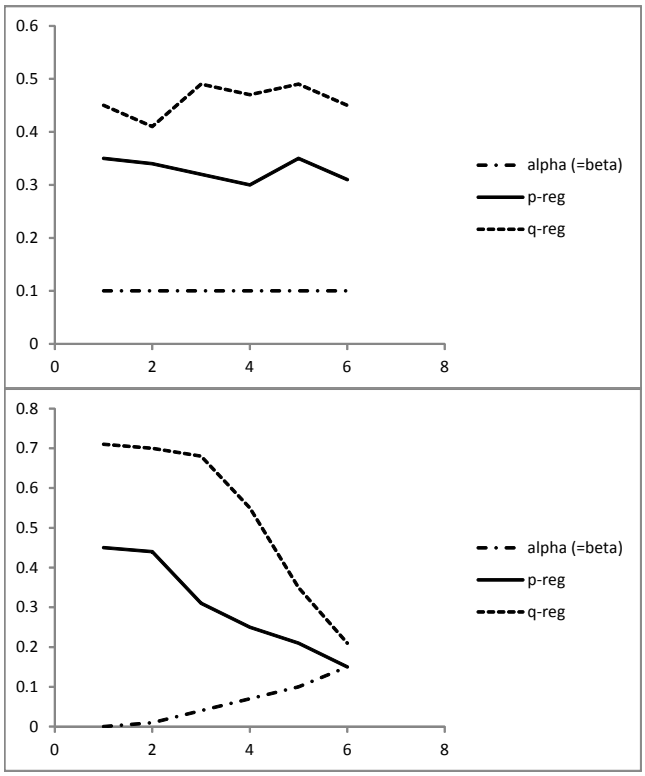

Figure 7. Graphs of regularization components values changes with the constant (top) and monotonous (below) increase of the regularization coefficients proportion. Labels at graphs: *p-reg* means $L_2(\alpha, \overline{w})$ values, *q-reg* means $L_3(\beta, \overline{w})$ values, *alpha(=beta)* – values of coefficients α, β ($\alpha = \beta$).

increasing value from a layer to a layer. The figures show that in case of the monotonic increase of coefficients of regularization components it is possible to minimize the values of the regularization components, i.e. implement the desired separation of the hidden representation on the p- and q-components.

In addition to assessing the separation quality using regularization term values, the visual assessment was made using examples from the test dataset. For this purpose, different images with different classes and transformations were passed through the network and then values of the hidden layer (of the encoder) were recombined. The p-components of the first image were "crossed" with the q-components of the second image. For this purpose, the trained autoencoder was divided into two parts – the encoding and decoding ones. The input image was "passed" through the encoding part and vector representation consisting of p- and q-components was taken from the encoder output. Then experimental software allowed recombining p- and q-components (i.e. take p-components from one image and join them with q-components from another image). After recombination, the new representation vector was passed through the decoder, providing a reconstructed image of the synthetic (recombinant) image at the output. Figure 8 demonstrates the examples where the q-components taken from other examples of distorted objects changed the appearance (transforms) of original image while keeping the class membership information.

Despite some noticeable artefacts, it can be concluded that we received qualitative experimental confirmation of the feasibility to separate class membership features from transform features, which motivates the further research in this direction.

3.1 Benchmark Dataset Results

MNIST-rot dataset was chosen to perform comparisons. MNIST-rot is a modified version of the original MNIST[27] dataset, often used by many researchers to compare effects of various regularizations[28,29]. MNIST-rot is a perfect dataset for investigating the properties of the suggested approach. The objects in this dataset are the handwritten digits under different rotation transformation which typically present difficulties to the modern autoencoder-based frameworks as the accuracy drops from the typical 97%-99% for the original MNIST dataset range to 70%-88% for MNIST-rot. We compared our framework against several state-of-the-art approaches of contracting

"A" image, the source of the p-component (class)	"B" image, the source of the q-component (transform)	Synthesized image, using the p-component of the "A" image and the q-component of the "B" image
 Number "5" on a black background shifted up and right	 Number "6" on a white background shifted down	 The image resembling number "5" on a white background, shifted down
 Number "2" on a black background shifted up and left	 Number "4" of smaller scale shifted up	 Number "2" of smaller scale on a black background shifted up
 Number "0" on a white background shifted up	 Number "1" on a black background shifted right	 The image resembling number "0" on a black background, shifted right

Figure 8. Examples of the p- and q- components recombination for the image representation.

autoencoders, denoising autoencoders and their variations (AE, CAE, DAE, mLDAE and mDAE). We've chosen the EIAE abbreviation, that stands for "explicit invariance autoencoder", in order to denote approach proposed. Comparison results are presented in Table 1.

The reason why we think EAI outperforms other methods is that it forces network to increase variety of features and explicitly requires that some of the features represent class-invariant characteristics while other represent the aspects of the transform, which it turn helps to distinguish the rotated figures from each other on the following classification and fine-tuning stages. For comparison, Figure 9 represents two sets of 400 features received with sAE (sparsity constrained autoencoder) and EIAE (the proposed method).

The results for the state-of-the-art methods were taken from29, except the wdAE and sAE that were carried out independently. In all experiments, except our approach autoencoder had 1000 neurons in the hidden layer, which then was concatenated with the classifier layer with fine-tuning. In our method just 400 neurons were enough to get maximum possible quality level. It can be concluded that the proposed approach is comparable with or even slightly outperforms the state-of-the-art methods as it requires fewer neurons to achieve about same level of quality. Further research should study the multilayer con-

Table 1. Error rate on MNIST-rot dataset

Method	Main idea behind the approach	Error rate, %
Baseline	Base SVM classifier on raw data	49.34
AE	Autoencoder without regularization	33.05
wdAE	AE + weight decay regularization	32.11
mLDAE	Marginalized linear autoencoder	25.31
sAE	AE + sparsity constraint regularization	18.81
CAE	Contractive auto-encoder	14.58
DAE	Denoising auto-encoder	12.61
mDAE	Marginalized denoising autoencoder	12.05
EIAE (ours)	Explicit invariance autoencoder	**11.94**

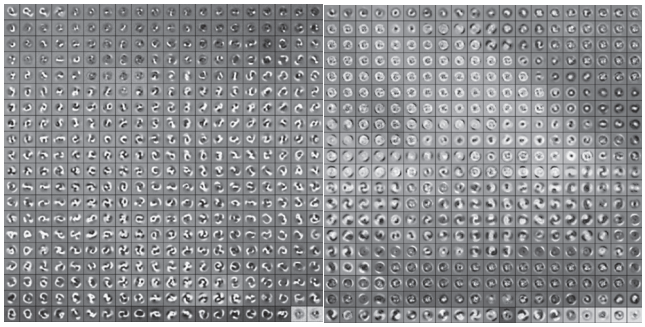

Figure 9. .Features of sAE (left) and EIAE (right) shown in the order of "mutual similarity". It is interesting to see many circle-like features of EIAE that were formed under the rotation invariance constraint. In contrast, sAE seems to memorize various digits in various positions.

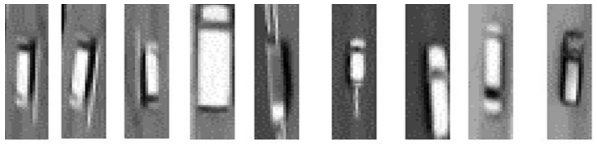

Figure 10. Sample images of vehicles under different transformations.

figurations for EIAE that should make it possible to obtain more class invariant features and thus further increase the accuracy of recognition.

4. Discussion

One of the key challenges of the pattern recognition is to design the transform-invariant feature extractor. In some practical applications it is also useful to know the parameters of the transformations that were applied to the pattern.

In this paper we presented a new regularization for Deep Learning Neural Networks (DLNN) capable of separating the hidden layer representation of the input patterns into two distinct groups: the features responsible for class membership information (invariant to transforms) and features responsible for transform representation. Such approach allows designing the image recognition systems invariant to translation, rotation and other transforms. Experiments with synthetic and benchmark datasets demonstrated the perspectives of this approach.

5. Conclusion

As for future work, a few issues must be resolved. Experiments demonstrate significant sensitivity to the regularization coefficients and high dependency on the regularization coefficients. So, it is necessary to develop a methodology for selecting optimal "weight" values for the regularization coefficients and to explore the possibilities of reducing dimensions of the p- and q-components from a layer to a layer, selecting the optimal dimension decrease rate, and explore the optimal ratio between the number of p- and q-components. Further investigation of this approach will be focused on the development of various kinds of transformations similarity estimation function ω to address the specific applications such as UAV image recognition. It is promising to research the possibility of using the proposed approach to eliminate image distortions of known origin and receive synthetic views of objects that were not available in the training dataset (for example, synthesizing the seasonal views of landscapes). An important future area of application of this method is analysis of areal images received from UAV, where it is important to determine object class irrespective of its transform – lighting, shadows, orientation, clutter, but at the same time the orientation of the object is also important (See Figure 10).

6. Acknowledgments

Part of the research was carried out with financial support from the Ministry of Education and Science under the grant agreement No. 14.576.21.0051 as of September 08, 2014 (agreement unique identifier RFMEFI57614X0051) to perform the applied research on the topic: "Development of intelligent algorithms of traffic situations detection and identification for the on-board systems of the unmanned aerial vehicles performing automatic traffic patrolling using GLONASS".

7. References

1. Deng L, Yu D. Deep Learning: Methods and Applications, Foundations and Trends® in Signal Processing: 2014; 7(3–4):197-387. Available from: http://dx.doi.org/10.1561/2000000039.
2. Rainer L, Maydt J. An extended set of Haar-like features for rapid object detection. IEEE, Proceedings of International Conference on Image Processing. 2002; 1:I-900-I-903. DOI: 10.1109/ICIP.2002.1038171.
3. Lowe DG. Object recognition from local scale-invariant features. Proceedings of the 7th IEEE international conference on Computer Vision, IEEE. 1999; 2:1150-57. DOI: 10.1109/ICCV.1999.790410
4. Bay H, Tuytelaars T, Van Gool L. Speeded-up robust features (SURF). Computer Vision and Image Understanding. 2008; 110(3):346-59. DOI: 10.1016/j.cviu.2007.09.014.
5. Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; 24(7):971-87. DOI: 10.1109/TPAMI.2002.1017623.
6. Dalal N, Triggs B. Histograms of oriented gradients for human detection. San Diego, United States: International Conference on Computer Vision & Pattern Recognition (CVPR '05), Jun 2005. IEEE Computer Society. 2005; 1:886-93.

7. Matas J, Chum O, Urban M, Pajdla T. Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing. 2004; 22(10):761-67.
8. Tuytelaars T, Mikolajczyk K. (2008) Local Invariant Feature Detectors: A Survey. Found. Trends. Comput. Graph. Vis. 2008 July; 3(3):177-280. DOI: 10.1561/0600000017.
9. Bishop CM. (1995). New York: Oxford University Press, Inc.: Neural networks for pattern recognition. 1995. ISBN:0198538642.
10. Szegedy C, Toshev A, Erhan D. Deep Neural Networks for Object Detection. Advances in Neural Information Processing Systems. 2013; 26:2553-61.
11. Kramer MA. Nonlinear principal component analysis using auto associative neural networks. AIChE Journal. 1991; 37(2):233-43. DOI: 10.1002/aic.690370209.
12. Krogh A, Hertz JA. A Simple Weight Decay Can Improve Generalization. Moody JE, Hanson SJ, Lippmann RP, (Eds.). San Mateo: Advances in Neural Information Processing Systems. 1992; 4:950-57.
13. Ranzato M'A, Boureau Y-L, LeCun Y. Sparse feature learning for deep belief networks. Advances in Neural Information Processing Systems. 2007; 20:1185-92.
14. Hanlin G, Thome N, Cord M, Lim J-H. Top-Down Regularization of Deep Belief Networks. Advances in Neural Information Processing Systems. 2013; 26:1878-86.
15. Hinton G. A practical guide to training restricted Boltzmann machines. Momentum. 2010; 9(1):926-43.
16. Hanlin G, Kusmierz L, Lim J-H, Thome N, Cord M. Learning Invariant Color Features with Sparse Topographic Restricted Boltzmann Machines. Belgium: Proceedings of 18th IEEE International Conference on Image Processing. 2011; p. 1241-44. DOI: 10.1109/ICIP.2011.6115657.
17. Yu K, Xu W, Gong Y. (2009) Deep learning with kernel regularization for visual recognition. Advances in Neural Information Processing Systems. 2008; 1889-96.
18. Rifai S, Vincent P, Muller X, Glorot X, Bengio Y. Contractive auto-encoders: Explicit invariance during feature extraction. Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011. Date accessed: 17/03/2015: Available from: http://www.icml-2011.org/papers/455_icmlpaper.pdf.
19. Tahri O, Chaumette F. Complex objects pose estimation based on image moment invariants. Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA). 2005; p.438-43.
20. Tran DT, Lee J-H. A Robust Method for Head Orientation Estimation Using Histogram of Oriented Gradients. Proceedings of the International Conference Signal Processing, Image Processing and Pattern Recognition. Communications in Computer and Information Science. 2011; 260:391-400. DOI: 10.1007/978-3-642-27183-0_41.
21. Chen J, Lai J, Feng G. Gabor-Based Kernel Fisher Discriminant Analysis for Pose Discrimination. Springer, Berlin-Heidelberg: Advances in Biometric Person Authentication. 2005; 3338:153-61. DOI: 10.1007/978-3-540-30548-4_18.
22. Kouskouridas R, Gasteratos A, Emmanouilidis C. Efficient representation and feature extraction for neural network-based 3D object pose estimation. Neurocomputing. 2013; 120:90-100. DOI: 10.1016/j.neucom.2012.11.047.
23. Kouskouridas R, Gasteratos A. Establishing low dimensional manifolds for 3D object pose estimation. IEEE International Conference on Imaging Systems and Techniques (IST). 2012; p. 425-30. DOI: 10.1109/IST.2012.6295483.
24. Rui N, Ji G, Zhao W, Feng C. ANN hybrid ensemble learning strategy in 3D object recognition and pose estimation based on similarity. Springer, Berlin-Heidelberg: Advances in Intelligent Computing. 2005; 3644:650-60. DOI: 10.1007/11538059_68.
25. Wunsch P, Winkler S, Hirzinger G. Real-time pose estimation of 3D objects from camera images using neural networks. Proceedings of the IEEE International Conference on Robotics and Automation. 1997; 4:3232-37. DOI: 10.1109/ROBOT.1997.606781.
26. Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Proceedings of IEEE International Conference on Neural Networks, IEEE. 1993; 586-91. DOI: 10.1109/ICNN.1993.298623.
27. LeCun Y, Cortes C, Burges C. The MNIST handwritten digit database. 1998. Date accessed: 27/05/2016: Available from: http://yann.lecun.com/exdb/mnist/.
28. Chen M, Weinberger KQ, Xu Zh, Sha F, Bengio Y. Marginalized denoising auto-encoders for nonlinear representations. Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014; 32:1476-84.
29. Chen, F-Q, Wu Y, Guo-Dong Zhao G-D, Zhang J-M, Zhu M, Bai J. Contractive De-noising Auto-Encoder. Springer International Publishing: Intelligent Computing Theory. 2014; 8588:776-81. DOI: 10.1007/978-3-319-09333-8_84.