CONFIDENTIAL EXTENDED ABSTRACT.
DO NOT DISTRIBUTE ANYWHERE.

Segmentation based on Transform Invariant Auto-encoder

Tadashi Matsuo[1,a)] Nobutaka Shimada[1]

1. Introduction

The auto-encoder method [1], [2], [4] is a type of unsupervised dimensionality reduction method. When encoding images by the auto-encoder method, a descriptor of an image generally differs from that of a spatially shifted version of the image as shown in Fig. 1 because a pattern itself and its position are inseparably embedded into a descriptor. Although the denoising auto-encoder method [6] can extract desired components from an input including information to be ignored, it requires an ideal output for each training sample when training an auto-encoder. Therefore, we need to normalize a position of an image to represent a pattern itself by such a descriptor. However, it is difficult to determine a standard position for normalizing images with complex boundaries such as hand images. In addition, if an image includes an unknown background, its descriptor may not accurately represent the image. We propose a transform invariant auto-encoder that outputs a descriptor invariant to some transforms such as shift and change of background. In this publication, by using the transform invariant auto-encoder, we generate a descriptor that represents a pattern itself without positional normalization even if target images include shifted variations and unknown backgrounds. By applying the proposed method to spatial shifts, we can generate an auto-encoder that separately extracts a descriptor of a pattern and a position of the pattern, where the descriptor is independent of shifts and it represents the pattern itself. By applying the proposed method to transforms that change contents of a background region, we can generate an auto-encoder that extracts a descriptor of a foreground region and a mask of the region from an image including an unknown background.

In this publication, we generate a shift invariant and background invariant auto-encoder as an example of the transform invariant auto-encoder. By some experiments, we demonstrate that a descriptor by the auto-encoder represents a pattern independent of its position and a background.

2. Ordinary auto-encoder

In general, an auto-encoder is so trained that the encoder–decoder combination approximately restores an input in a certain input set. It is formulated as a problem minimizing a cost function $C_{\mathrm{ord}}(E, D)$ defined as

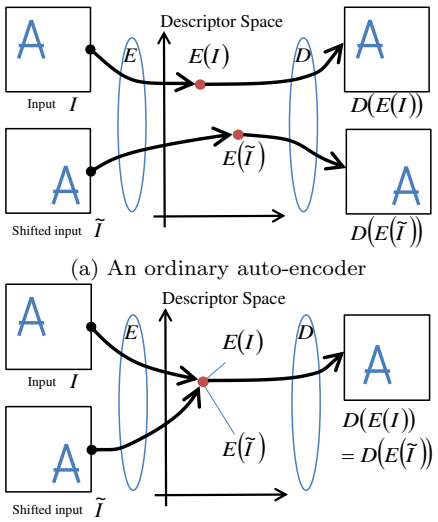
$$C_{\mathrm{ord}}(E, D) = \sum_{I \in S} \|I - D(E(I))\|_2^2, \tag{1}$$

where S, $E(\cdot)$, $D(\cdot)$, and $\|\cdot\|_p$ denote a set of inputs, the encoder, the decoder, and the ℓ^p norm, respectively.

To minimize $C_{\mathrm{ord}}(E, D)$, the decoder should be able to approximately restore an original vector I from its descriptor $E(I)$, which has a lower dimensionality than I. By training the encoder E and the decoder D by minimizing $C_{\mathrm{ord}}(E, D)$, information sufficient to restore an original vector can be extracted as a descriptor by the encoder. In this way, the auto-encoder method can construct descriptors of vectors from just a set of training vectors.

However, a descriptor of an image from an ordinary auto-encoder includes both a spatial pattern and its position.

If images have a common spatial pattern at different positions, their descriptors are different.



(a) An ordinary auto-encoder

(b) A shift invariant auto-encoder

Fig. 1 Characteristics of auto-encoders

3. Transform invariant auto-encoder

As a method to construct a descriptor representing a spatial pattern, we propose a transform invariant auto-encoder. With the proposed method, close descriptors are allocated to two inputs if they are mapped to each

[1] Ritsumeikan University
[a)] matsuo@i.ci.ritsumei.ac.jp

other by a certain set of transforms. We call the set "ignored transforms". A transform invariant auto-encoder is generated by training an auto-encoder with a novel cost function. The cost function should induce the accurate restoration of a pattern as well as transform invariance. We achieve such an cost function by adding a transform variance term and relaxing the restoration error term.

3.1 Transform variance term

As a measure of the transform variance, we propose a sum of differences between a restored image and an image restored from a transformed input as follows:

$$C_{\mathrm{inv}}(E,D) \stackrel{\mathrm{def}}{=} \sum_{I\in S}\sum_{i} \left\| D\left(E\left(I\right)\right) - D\left(E\left(T_{\theta_i}\left(I\right)\right)\right)\right\|_2^2, \tag{2}$$

where S and T_θ denote a set of training inputs and a transform operator in the ignored transforms, respectively. To minimize (2), the combination of the encoder E and the decoder D need to output similar vectors for variously transformed versions of an input. By optimizing the encoder E and the decoder D so that they minimize (2), their combination is approximately transform invariant for inputs in the set S.

3.2 Restoration error term

To compare patterns without respect to ignored transforms, we need to relax the restoration error cost in (1) so that it will be small if a restored input matches a transformed version of its original input. Therefore, we propose the following term as a measure of the accuracy of the restoration of a pattern:

$$C_{\mathrm{res}}(E,D) \stackrel{\mathrm{def}}{=} \sum_{I\in S} \min_{i} \left\| T_{\theta_i}\left(I\right) - D\left(E\left(I\right)\right)\right\|_2^2. \tag{3}$$

To minimize (3), the restored image $D(E(I))$ should approximately match one of the transformed inputs $\{T_{\theta_i}(I)\}$. This means that the pattern should be approximately restored.

3.3 Cost function

Our total cost function $C(E,D)$ is formulated as follows;

$$\begin{aligned} C\left(E,D\right) \stackrel{\mathrm{def}}{=} & \lambda_{\mathrm{inv}} C_{\mathrm{inv}}\left(E,D\right) + \lambda_{\mathrm{res}} C_{\mathrm{res}}\left(E,D\right) \\ & + \lambda_{\mathrm{spa}} \sum_{I\in S} \left(\frac{\left\|E\left(I\right)\right\|_1}{\left\|E\left(I\right)\right\|_2} \right)^2, \end{aligned} \tag{4}$$

where λ_{inv}, λ_{res}, and λ_{spa} denote the scalar weights of each term. The third spatial sparseness term causes that similar inputs are encoded into descriptors close to each other[5]. We train the encoder E and the decoder D so that they minimize the proposed cost function $C\left(E,D\right)$.

4. Inference of transform parameter

We propose an inference method of a transform parameter which is ignored by a transform invariant auto-encoder. We define a transform parameter of an input I as a parameter representing a transform from the input I to the restored input $D(E(I))$. For example, a transform parameter for a shift invariant auto-encoder means a spatial shift.

An input can be approximately restored from its descriptor and transform parameter. Therefore, a pair of a transform invariant auto-encoder and the corresponding inference model of a transform parameter is an auto-encoder that can represent an input as a pair of a transform invariant part and a transform variant part.

We propose the following cost function to train an inference model R of a transform parameter.

$$C_{\mathrm{par}}(R) = \sum_{I\in S} \left\| R\left(I\right) - \operatorname*{argmin}_{\theta} \left\| I - T_\theta\left(D\left(E\left(I\right)\right)\right)\right\|_2^2 \right\|_2^2. \tag{5}$$

We can achieve an inference model R of a transform parameter by minimizing $C_{\mathrm{par}}(R)$.

5. Segmentation based on transform invariant auto-encoder

We apply the proposed transform invariant auto-encoder to segmentation of a foreground from an image with a complex background. Since a foreground is invariant with respect to transforms that change contents of a background region, we can generate a transform invariant auto-encoder that extracts a foreground from an image.

In this case, there exist extremely many transforms to be ignored. Since the invariance term C_{inv} in (2) means the constraint that the encoder and decoder should be invariant with respect to a set of transforms, the effect of the term can be approximated by calculating the term with a subset of ignored transforms. Here, we calculate the invariance term C_{inv} with transforms randomly selected for each sample in each training step.

On the other hand, the restoration term C_{res} in (3) means the constraint that a vector restored from an input should be close to one of vectors transformed from the input. For convergence of training, the set of the transformed vectors should be fixed for each input. In addition, vectors restored from similar inputs should be compared with similar transformed vectors. Therefore, instead of randomly selected transforms, we calculate the restoration term with a representative transform that changes values on the background to 0.

Finally, we propose the following cost function for segmentation.

$$\begin{aligned} & C_{\mathrm{seg}}\left(E,D\right) \\ \stackrel{\mathrm{def}}{=} & \sum_{I\in S} \lambda_{\mathrm{inv}} \sum_{\mathrm{random}\,\theta} \left\| D\left(E\left(I\right)\right) - D\left(E\left(T_\theta\left(I\right)\right)\right)\right\|_2^2 \\ & + \lambda_{\mathrm{res}} \left\| \tilde{T}_0\left(I\right) - D\left(E\left(I\right)\right)\right\|_2^2 \\ & + \lambda_{\mathrm{spa}} \sum_{I\in S} \left(\frac{\left\|E\left(I\right)\right\|_1}{\left\|E\left(I\right)\right\|_2} \right)^2, \end{aligned} \tag{6}$$

where $\tilde{T}_0$ means the transform that changes values on the background to 0.

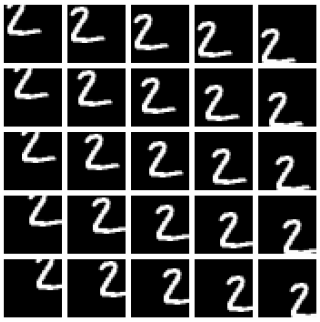


Fig. 2 An image in MNIST and its shifted versions

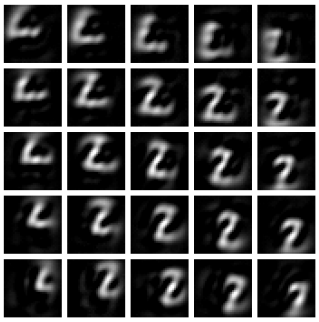


Fig. 3 Images restored using an ordinary auto-encoder

Fig. 4 Images restored using a shift invariant auto-encoder

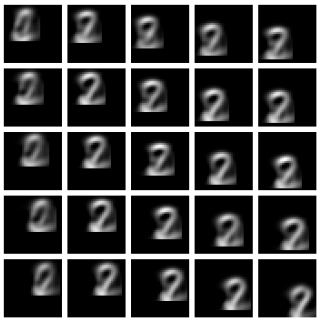


Fig. 5 Images restored using a shift invariant auto-encoder with inferred shifts

6. Experiments

6.1 Experiments for pattern representation

We demonstrate the effectiveness of the proposed method by experiments with a shift invariant auto-encoder. The shift operator T_{θ_i} is defined as

$$\left(T_{\theta_i}\left(I\right)\right)\left(x,y\right) = I(x+\Delta x_i, y+\Delta y_i), \tag{7}$$

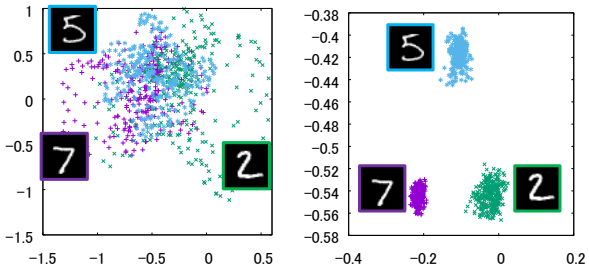
where $I(x,y)$ denotes the value of the image I at the position (x,y). We used the following shift parameters:

$$\{(\Delta x_i, \Delta y_i)\} = \{-8,-6,-4,-2,0,2,4,6,8\}^2. \tag{8}$$

Here, we demonstrate shift invariant property of the proposed method using experiments for digit patterns.

As an encoder, we used a neural network consisting of a single convolutional neural network (CNN)[3] following a three-layer fully connected neural network (NN). As a decoder, we used a three-layer fully connected NN. In addition, we used a hyperbolic tangent as an activation function, which is placed between each pair of layers. We generated two pairs of encoders and decoders with the same structure. One was trained as an ordinary auto-encoder by minimizing (1), and the other was trained as a shift invariant auto-encoder by minimizing (4) for digit images in the MNIST database [3]. For the ordinary auto-encoder, we used additional images that were randomly shifted according to the parameters in (8). Both auto-encoders were trained by stochastic gradient descent (SGD) [3], and both were updated with every 50 samples that were randomly extracted from the MNIST database. We used auto-encoders that were updated 100,000 times. We also trained an inference model R of a shift parameter. The inference model consisted of a three-layer fully connected NN.

As an example, we encoded and decoded an image of the digit "2", which is not used in training auto-encoders. Input images are shown in Fig. 2, where the center image is the original image in the MNIST database and the others are its shifted versions. Images in Fig. 3 are restored from images in Fig. 2 using an ordinary auto-encoder. Images restored by a proposed shift invariant auto-encoder are shown in Fig. 4. Fig. 5 shows the restored images which are shifted according to the shift parameters estimated by the inference model R. In Fig. 3, the restored



(a)by an ordinary auto-encoder (b)by an shift invariant auto-encoder

Fig. 6 The distribution of descriptors of shifted images

images are located depending on the shifts in the input images. Conversely, the restored images in Fig. 4 are very similar to each other. This means that the proposed shift invariant auto-encoder can encode an input image of the digit "2" into a descriptor representing the typical spatial shape of the digit "2" without regard to shifts in the input image.

In addition, we calculated the distributions of the descriptors from the shifted images. We encoded the digit images corresponding to "2", "5", and "7" and their shifted versions using the two auto-encoders. Fig. 6(a) shows the distributions from the ordinary auto-encoder, and Fig. 6(b) shows those from the shift invariant auto-encoder. In these figures, 30 dimensional descriptors are projected onto a two-dimensional space spanned by the three mean vectors of the descriptors for each digit. By comparing these figures, we see that descriptors generated by the shift invariant auto-encoder are obviously concentrated for each digit. With a shift invariant auto-encoder, descriptors from shifted images of the same digit are close to each other and descriptors from shifted images of other digits are far from each other. This means that a descriptor generated by a shift invariant auto-encoder represents the spatial pattern. In addition, descriptors in Fig. 6(b) make clusters corresponding to digits, even though we have entered no digit information when training the shift invariant auto-encoder. The proposed method may be applicable to the unsupervised clustering of images based on their spatial patterns.

6.2 Experiments for segmentation

We demonstrate an auto-encoder that extracts a fore-

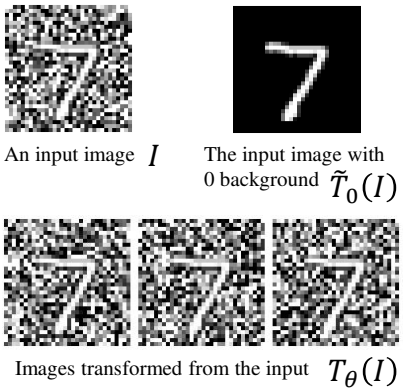


Fig. 7 A training sample and transformed images for segmentation

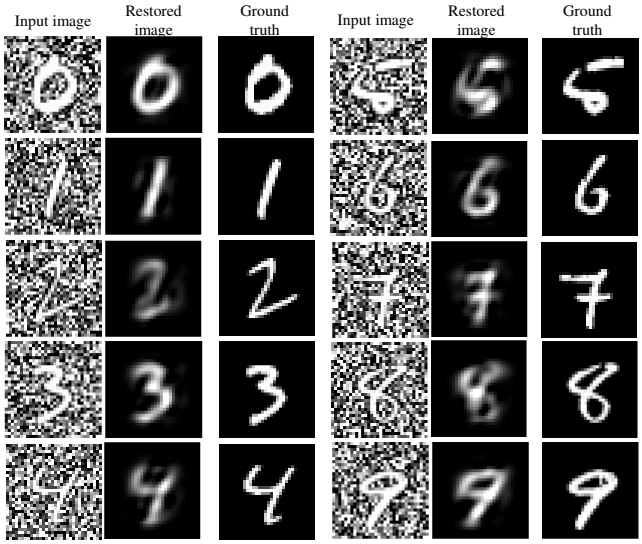


Fig. 8 Segmentation of images with noisy background

ground from a MNIST image with a noisy background. We generated the training samples by adding a random background generated by Gaussian white noise, where the values on the foreground is near to 1, the mean of the noise is 0.5 and the standard deviation of the noise is also 0.5. In Fig. 7, we show a training sample and transformed images used in (6). We trained an auto-encoder with the structure similar to that in 6.1 so that it minimizes the proposed cost function (6).

After training the auto-encoder, we encoded some images not used in the training process and restore images by the decoder. The restored images are displayed in Fig. 8.

We also generated another auto-encoder trained with images with backgrounds randomly selected from indoor scenes. When generating training samples, we converted values on a foreground from 1 to 0.5 so that the foreground cannot be easily segmented by absolute values of pixels. In Fig. 9, we display images restored from those not used in the training process. Although details of the digits "2" and "5" are not so accurately restored, their whole forms are roughly restored. The forms of the other digits are approximately segmented from images without regard to their backgrounds.

These results show that the auto-encoder successfully

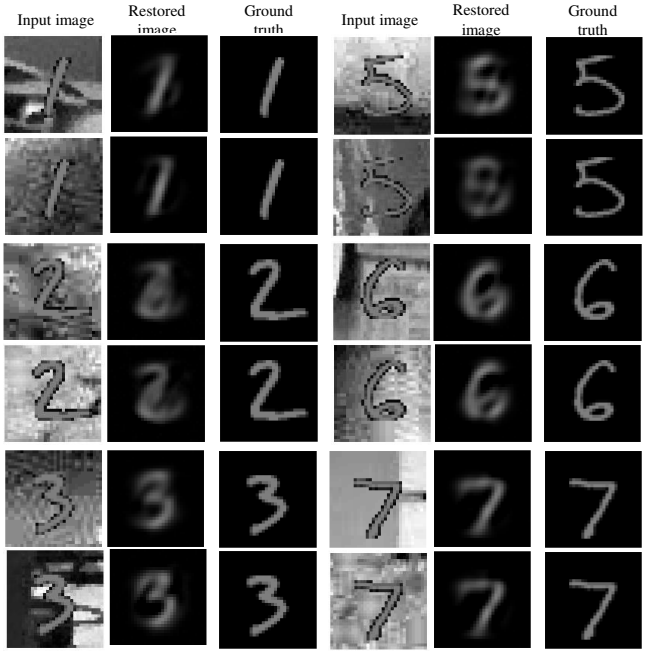


Fig. 9 Segmentation of images with random scene background

segments approximate forms of digits as foregrounds.

7. Conclusion

We proposed a transform invariant auto-encoder and demonstrated a shift invariant auto-encoder and a background invariant auto-encoder. By several experiments, we showed that they can encode a pattern independently of its position and a background, respectively. By combining their cost function, we will be able to generate an auto-encoder that encodes a typical foreground without regard to its position and background.

References

[1] Baldi, P. and Hornik, K.: Neural networks and principal component analysis: Learning from examples without local minima, *Neural Networks*, Vol. 2, No. 1, pp. 53 – 58 (1989).

[2] Hinton, G. E. and Salakhutdinov, R. R.: Reducing the Dimensionality of Data with Neural Networks, *Science*, Vol. 313, No. 5786, pp. 504–507 (2006).

[3] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324 (1998).

[4] Makhzani, A. and Frey, B. J.: k-Sparse Autoencoders, *CoRR*, Vol. abs/1312.5663 (2013).

[5] Matsuo, T. and Shimada, N.: Construction of Latent Descriptor Space of Hand-Object Interaction, *Proceedings of 22nd Japan-Korea Joint Workshop on Frontiers*, pp. 117–122 (2016).

[6] Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.-A.: Extracting and Composing Robust Features with Denoising Autoencoders, *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, New York, NY, USA, ACM, pp. 1096–1103 (2008).