Learning Transformation Invariance from Global to Local

Jens Hocke, Thomas Martinetz

Institute for Neuro- and Bioinformatics, University of Lübeck

Abstract. Learning representations invariant to image transformations is fundamental to improving object recognition. We explore the connections between i-theory, Toroidal Subspace Analysis and slow subspace learning. All these methods can only achieve invariance to one transformation. Motivated by this limitation of these global methods we adapt the slow subspace approach to a local convolutional setting. Experimentally we show invariance to multiple transformations, and test object recognition performance.

1 Introduction

Changes of an object's pose are one of the big challenges in visual object recognition. The pixel representation of an object can change dramatically when the object's pose changes. Often this problem is met by presenting many training examples of the object in different poses. However, to achieve human like capability to learn from few samples it seems mandatory to separate the invariance learning from the object recognition problem.

Convolutional neural networks [1, 2] are an early example of an architecture that helps coping with shift invariance. Theses convolutional networks do not come with an objective function to learn invariance. Their main goal is classification, and invariance is learned as part of classification.

A well suited objective to achieve an invariant representation is slowness [3–5]. The main assumption of slowness is, that there is some slowly changing signal contained in a temporal stream of data. By optimizing for a slowly changing signal in the representation, invariance to the transformations contained in that temporal stream is achieved.

Pairs of filters coupled by their energy, so called subspaces, have shown to be a very useful architecture. The subspaces have been introduced in the domain of self organizing maps [6], and have been transferred to the representation learning domain in form of the Independent Subspace Analysis (ISA) [7]. Soon slowness and subspace architectures were combined in [8], minimizing the energy change of the subspaces over time. Newer approaches [9–11] also include sparsity [12].

An approach derived from group theory is the Toroidal Subspace Analysis (TSA) [13]. The resulting representation also uses subspaces. In contrast to the slowness based subspace approaches, the energy of the subspaces is fixed for pairs of transformed image patches, and the error for encoding one patch in terms of the other is minimized. Similar to TSA, gated models [14] minimize the encoding distance between pairs of transformed images. However, there products of filters are used to encode the transformations.

An explanation of how invariance could emerge in the ventral stream is offered in the i-theory [15]. From these theoretical insights on invariance, implications for the network structure can be derived.

After explaining the connections between i-theory, TSA and slow subspace learning methods and their drawbacks, we adapt the local subspace learning method by W. Zou et al. [9, 10] to convolutional learning and test this method for invariance and unique representation.

2 Transformation Groups and Invariance

Orbits can be used to achieve invariance to a group G of transformations. This is the core observation of the i-theory [15] as well as integral invariants. The group elements g are transformations of images $\boldsymbol{x} \in \mathbb{R}^D$. We denote the group's actions on an image by $g(\boldsymbol{x})$. The orbit $O_{\boldsymbol{x}} = \{g_i(\boldsymbol{x}) | g_i \in G\}$ of some image $\boldsymbol{x}$ is induced by applying all transformations $g_i \in G$ to $\boldsymbol{x}$. This orbit is invariant to the transformations in G and unique for the object in $\boldsymbol{x}$. But it is a very high dimensional representation.

The high dimensionality can be handled by one dimensional projections $\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle$, where $\boldsymbol{p}_n, n = 1, \dots D$ are arbitrary projection vectors. If there are enough different projection vectors, a unique representation can be achieved. Besides reducing the dimensionality, this helps to avoid transforming the input image $\boldsymbol{x}$ by applying the inverse transformation to the projection vectors instead

$$\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle = \langle \boldsymbol{x}, g_i^{-1}(\boldsymbol{p}_n) \rangle. \tag{1}$$

So now we have projected orbits of images. In the i-theory probability distributions over these vectors are used to obtain an invariant representation. This helps analyzing the invariance problem. However, we found it hard to learn good representations using this probabilistic framework [16]. Therefore, we stay in the deterministic domain.

If we assume that the transformations g_r have only one parameter r (e.g. degree for rotation) and they are ordered by this parameter, we can assemble a matrix W. This matrix $W = (g_{r_1}^{-1}(\boldsymbol{p}), g_{r_2}^{-1}(\boldsymbol{p}), \dots, g_{r_N}^{-1}(\boldsymbol{p}))$ is composed of column vectors $\boldsymbol{w}_r = g_r^{-1}(\boldsymbol{p})$. The parameters r_i for the transformations are uniformly distributed $r_i = N/I \cdot (i-1)$ with I being the maximum transformation parameter. In the following line of thinking, we assume one projection vector $\boldsymbol{p}$. Here, we abbreviate g_{r_i} by g_i and w_{r_i} by w_i. The representation $\boldsymbol{y}$ of the image vector $\boldsymbol{x}$ is obtained by

$$\boldsymbol{y} = W^\top \boldsymbol{x}. \tag{2}$$

For the transformed image $g_j(\boldsymbol{x})$ we obtain $\boldsymbol{y}'$. If we observe a single entry y_i of $\boldsymbol{y}$ while applying transformation g_j

$$\begin{aligned} y_i =& w_i^\top \boldsymbol{x} && (3)\\ =& g_j(w_i)^\top g_j(\boldsymbol{x}) && (4)\\ =& w_{i+j}^\top g_j(\boldsymbol{x}) = y'_{i+j}, && (5) \end{aligned}$$

we see that the entries of the representation vector shift indices. Only the first or last elements of $\boldsymbol{y}'$, depending on the direction of the transformation, may not be related to $\boldsymbol{y}$. By restricting the applicable transformations G to the set of toroidal group transformations, a relation to all entries in $\boldsymbol{y}$ can be established.

These toroidal group transformations are turned into circular shifts in the representation vector $\boldsymbol{y}$. So via the Fourier transform of $\boldsymbol{y}$ amplitudes invariant to toroidal group transformations can be found, while the phases encode the transformation parameter. Via the n-dimensional Fourier transform an extension to n parameters is possible.

2.1 Relationship of Invariance Learning Methods

In case the transformation group is not known or the transformation is hard to model, it is beneficial to learn W. Let $\boldsymbol{x}(t) = g(\boldsymbol{x}(t-1))$ at time t be a transformed version of an image $\boldsymbol{x}(t-1)$ in a sequence. From above we know that the Fourier amplitudes will not change. Only the phase will change according to the Fourier shift theorem. Thus, we can reconstruct $\boldsymbol{x}(t)$

$$\boldsymbol{x}(t) = W^{-1}F^{-1}R(\phi)FW\boldsymbol{x}(t-1) \tag{6}$$

if the phase shift ϕ encoded in a diagonal matrix $R(\phi)$ is known. This can be turned in a learning algorithm, where this autoencoder like energy term

$$E = \sum_t ||\boldsymbol{x}(t) - W^{-1}F^{-1}R(\phi)FW\boldsymbol{x}(t-1)||, \tag{7}$$

and $R(\phi)$ are optimized in an alternating manner. Since the Fourier transformation is just an unitary transformation, it can be absorbed into W

$$E = \sum_t ||\boldsymbol{x}(t) - W^\top R(\phi)Wx(t-1)||. \tag{8}$$

This is the essence of TSA [13], where usually the complex unit vectors on the diagonal of $R(\boldsymbol{\phi})$ are not coupled.

Related to TSA are slow subspace approaches. They have two main ingredients. They encourage a representation that allows reconstruction of $\boldsymbol{x}(t)$ from $W^\top \boldsymbol{x}(t)$, which can be achieved via an orthogonal basis W, an autoencoder term or sparse coding. In order to find an invariant representation changes in the subspace energies

$$e_i(t) = \sum_{k=0}^{K-1} (\boldsymbol{w}_{iK+k}^\top \boldsymbol{x}(t))^2 \tag{9}$$

of K-dimensional subspaces indexed by i are penalized. This is done either by minimizing their distance in consecutive samples or by minimizing their variance[1], which also indirectly minimizes the subspace energy of samples following each other. In the following we assume $K = 2$.

The relation of subspace methods to TSA and the i-theory can be seen if all energy terms are zero for any pair of group transformed images. Then subspace methods have found a basis W_{slow}, that can reconstruct all sample pairs $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t-1)$ from a sequence, while $\boldsymbol{e}(t)$ does not change for consecutive samples. Only the pairs of activations $w_{i2}^{\top}\boldsymbol{x}(t)$ and $w_{i2+1}^{\top}\boldsymbol{x}(t)$ can change over time. This change can be interpreted as an angle change in polar coordinates, which is the only change TSA allows to reconstruct $\boldsymbol{x}(t)$ from $\boldsymbol{x}(t-1)$. From that, we see, any input image can be group transformed using W_{slow} and some matrix $R(\boldsymbol{\phi})$ for the angle change. Therefore, W_{slow} is also an optimal solution for the TSA model. The other way round, an optimal basis W_{TSA} learned by TSA, will always have a fixed $\boldsymbol{e}(t)$, and perfect self-reconstruction is guaranteed via not transformed pairs $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t-1)$. Thus, W_{TSA} is also an optimal solution for the subspace model.

2.2 A Slow Subspace Autoencoder Model

The model we chose to build on is a slow subspace autoencoder model [10], that has been applied successfully in object recognition tasks. It follows the scheme mentioned above. There is a reconstruction and a slowness term, and in addition also sparsity is encouraged. The terms

$$E_{rec} = \sum_t ||\boldsymbol{x}(t) - W^{\top}W\boldsymbol{x}(t)||_2^2 \tag{10}$$

$$E_{slow} = \sum_t ||\boldsymbol{z}(t) - \boldsymbol{z}(t-1)||_1 \tag{11}$$

$$E_{sparse} = \sum_t ||\boldsymbol{z}(t)||_1 \tag{12}$$

with the amplitudes

$$z_i(t) = \sqrt{e_i(t)} \tag{13}$$

are combined via

$$E = E_{rec} + \alpha E_{slow} + \beta E_{sparse} \text{ s.t. } ||\boldsymbol{w}_i|| = 1. \tag{14}$$

Note the unit norm constraint on the weight vectors $\boldsymbol{w}_i$. This is necessary to avoid $\boldsymbol{w}_i$ to become a zero vector if large values of α or β are used. This energy model can now be optimized via stochastic gradient descent.

[1] The principle of minimizing the variance over time is also fundamental to the Slow Feature Analysis [5]. However, SFA is not operating on subspaces.

In case the sparsity term is omitted TSA [13] like bases are found using the same training samples (Figure 1a and b). These bases are global and the amplitudes are perfectly transformation invariant. However, a large amount of spatial information is lost with the phases. This representation is sensible to background clutter and factorizing the invariance problem seems impossible (e.g. one module doing translation invariance, the next rotation etc.) [17]. But factorizing the range of possible transformation parameters is possible according to the i-theory [15]. Factorizing the invariance means we are restricting it to a local window, which can be achieved via a sparsity term (Figure 1c). Of course this will decrease the invariance [18], but due to the local similarity of most transformations to shifts, a diverse set of transformations can be handled. By adding additional layers trained like this first module, the range of invariance can be increased.

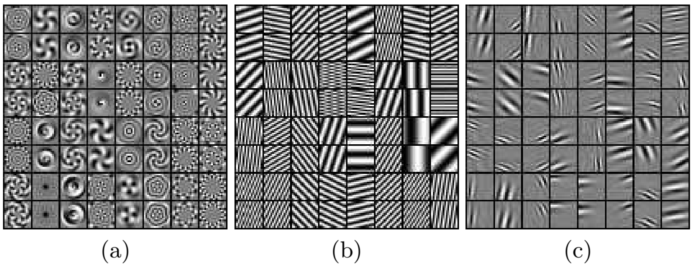


Fig. 1. The first 64 elements of global bases learned from rotated and shifted patches of random intensities are shown in (a) and (b). In (c) the first 64 elements of a basis learned with a sparsity prior from natural movie sequences is shown.

2.3 Convolutional Model

When the invariance is factorized to local windows by optimizing sparsity, many very similar shifted local subspaces are created (Figure 1c). We decrease this redundancy by training an adapted convolutional network with the energy terms

$$E_{rec} = \sum_t ||x(t) - \sum_j \left(\tilde{W}_j * u_s(d_s(W_j * x(t)))\right) ||_2^2 \tag{15}$$

$$E_{slow} = \sum_t ||d_s(\boldsymbol{z}(t)) - d_s(\boldsymbol{z}(t-1))||_1, \tag{16}$$

where

$$z_i(t) = \sqrt{\sum_{k=0}^{1} (W_{i2+k} * x(t))^2}. \tag{17}$$

Here, the vectors $\boldsymbol{w}$ are replaced by filters W_j and their counter parts with all dimensions flipped $\tilde{W}_j$. The convolution operation is denoted by $*$. In addition the downsampling and upsampling operators d_s and u_s with stride s are introduced, taking only every s-th value in each direction or reversing the downsampling by filling in zeros. Due to the network structure no sparsity needs to be enforced to learn local subspaces and the required computational resources stay moderate.

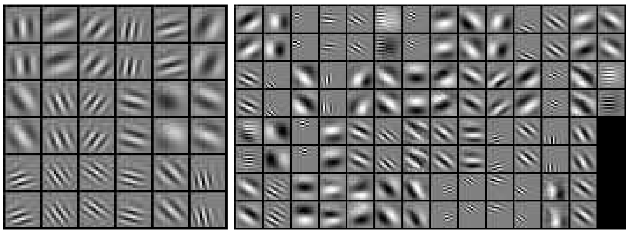
For training the first layer on sequence images, the images were preprocessed by ZCA filtering [19]. Using these preprocessed images, filters for the convolutional model were optimized by stochastic gradient descent. After training, the first layer output maps $d_s(z_i(t))$ can be computed via (17). The next layer is trained on the output of the first layer for unprocessed images. Because these outputs can be high dimensional, the number I of maps is reduced by filtering. As filters we use the principle components of $1 \times 1 \times I$ patches extracted from the outputs. This data is then ZCA filtered and used for optimizing the second layer filters. Higher layers can be computed analog to the second layer. For computing the outputs of higher layers, only the PCA step is needed, and thus, ZCA filtering is omitted.

3 Experiments

We trained a two layer version of the convolutional model using natural movie sequences from the van Hateren video database [20]. These are gray scale 128×128 pixel movies collected from television. For training the first layer with $\alpha = 50$, the stride was set to 6, the filter size was set to 15×15 pixels and 36 filters shown in Figure 2a were trained. Then using the learned filters, the first layer output was generated using stride 2. To train the second layer the 18 magnitude maps from the first layer output were reduced via PCA to three maps carrying more than 90% of the variance. Again for training the stride of the second layer was set to 6. The filter size was adapted to $15 \times 15 \times 3$ to handle all three maps and 108 filters were learned with $\alpha = 100$. We see the results for the top map in Figure 2b. Note, we did not analyze many of the parameters. One might find better choises. In particular, the second layer filters are critical and for many parameters no useful filters will be produced.

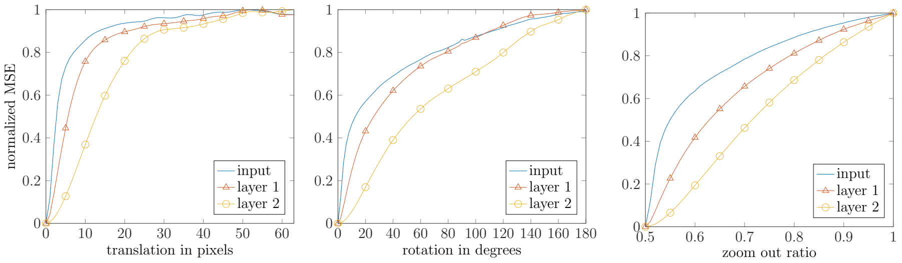
Clearly for both layers we obtain Gabor like filters (Figure 2). For the second layer the filters are repeated in every map, however with different intensities. These finding suggest invariance to small shifts in the first layer and an increased invariance to these shifts in the second layer. We tested this translation invariance and also rotation and scale invariance using 100 patches of 64×64 pixels from the van Hateren image database [21]. We measure the change in the output of each layer, as the input undergoes transformations. The MSE between the original and the transformed patch is taken and normalized against the largest MSE, assuming the patches are uncorrelated for these transformation parameters. The output of both layers were downsampled with stride 3.

The plots in Figure 3 validate our believe in invariance to small shifts. We also see invariance to rotation and scaling, because these transformations can locally be approximated by shifts. And additionally the invariance increases from the first to the second layer.



(a) (b)

Fig. 2. Layer 1 filters are displayed in (a). In (b) only the top part of the second layer filters is shown. This top part, which is for the first output map, differs from the other parts only in the intensities of the filters.



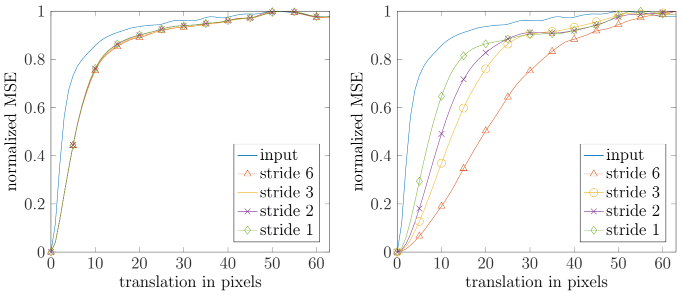
(a) (b) (c)

Fig. 3. Invariance experiment for varying degrees of shift (a), rotation (b), and scale (c). The normalized MSE in layer 1 and layer 2 is plotted along with results for the unprocessed input patches as reference.

Next we were interested in the effect of the stride. The strides for both layers were adapted simultaneously. Using the same approach as above we measured the MSE for different strides on shifted patches. The plots in Figure 4 show, that the first layer output is not affected. However, the second layer is. This is due to the change of the represented area. The larger the stride in the bottom layer the larger the area represented in the second layer.

These findings suggest using large strides. One of the main problems of invariant representations, however, is representing the input uniquely. To test how well information on fine image structures is retained at each layer we do k-NN classification ($k = 3$) on the MNIST [2] dataset. The classification error on the raw images is 3.09%. As we see in Table 1, there is a drop in the k-NN classification performance from layer 1 to layer 2, which can be reduced to a certain extend by choosing small stride sizes. This clearly indicates a loss of important information. Interestingly, the first layer error rates are significantly better than on the input images[2]. We think this is due to the small non-affine transformations in MNIST, which may be handled well by the Gabor features.

[2] The state of the art error rate for MNIST is of 0.23% [22].



(a)

(b)

Fig. 4. The normalized MSE in layer 1 (a) and layer 2 (b) depending on the amount of shift is plotted for different strides. As reference also a curve for the input patches is shown.

	Layer 1	Layer 2
Stride 6	1.48	12.88
Stride 3	**1.41**	6.35
Stride 2	1.43	5.01
Stride 1	1.42	**3.28**

Table 1. Results for MNIST classification. The error rates are given in percent.

4 Conclusion

I-theory, TSA and slow subspace learning methods are closely related. Invariance learning based on anyone of these seems equally well suited, leaving aside optimization and implementation issues. However, if they learn global invariance, their application is very limited by their adaption to a single transformation group. Therefore, we implemented a convolutional method, which learns local invariance due to its structure. The experiments show indeed invariance to multiple transformations, with increase in invariance from layer to layer, while information loss also seems to be increased. This information loss remains an open problem to be solved before deeper networks using our training approach become useful. Interestingly, the first layer seems to be capable of handling non-affine transformations in MNIST, leading to improved classification results.

References

1. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics **36** (1980) 193–202
2. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
3. Hinton, G.E.: Connectionist learning procedures. Artificial intelligence **40**(1) (1989) 185–234

4. Földiák, P.: Learning invariance from transformation sequences. Neural Computation **3**(2) (1991) 194–200
5. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: Unsupervised learning of invariances. Neural computation **14**(4) (2002) 715–770
6. Kohonen, T.: Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. Biological Cybernetics **75**(4) (1996) 281–291
7. Hyvärinen, A., Hoyer, P.: Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. Neural computation **12**(7) (2000) 1705–1720
8. Kayser, C., Einhäuser, W., Dümmer, O., König, P., Körding, K.: Extracting slow subspaces from natural videos leads to complex cells. In: Artificial Neural Networks—ICANN 2001. Springer (2001) 1075–1080
9. Zou, W.Y., Ng, A.Y., Yu, K.: Unsupervised learning of visual invariance with temporal coherence. In: NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning. (2011)
10. Zou, W., Zhu, S., Yu, K., Ng, A.Y.: Deep learning of invariant features via simulated fixations in video. In: Advances in Neural Information Processing Systems. (2012) 3212–3220
11. Cadieu, C.F., Olshausen, B.A.: Learning intermediate-level representations of form and motion from natural movies. Neural computation **24**(4) (2012) 827–866
12. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research **37**(23) (1997) 3311–3325
13. Cohen, T., Welling, M.: Learning the irreducible representations of commutative lie groups. arXiv preprint arXiv:1402.4437 (2014)
14. Memisevic, R.: Learning to relate images. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(8) (2013) 1829–1846
15. Anselmi, F., Leibo, J.Z., Rosasco, L., Mutch, J., Tacchetti, A., Poggio, T.: Unsupervised learning of invariant representations in hierarchical architectures. CoRR **abs/1311.4158** (2013)
16. Hocke, J., Martinetz, T.: Learning transformation invariance for object recognition. In: Workshop New Challenges in Neural Computation 2014. (2014) 20–25
17. Anselmi, F., Poggio, T.A.: Representation learning in sensory cortex: a theory. (2014)
18. Lies, J.P., Häfner, R.M., Bethge, M.: Slowness and sparseness have diverging effects on complex cell learning. PLoS computational biology **10**(3) (2014) e1003468
19. Bell, A.J., Sejnowski, T.J.: Edges are the" independent components" of natural scenes. In: NIPS. (1996) 831–837
20. van Hateren, J.H., Ruderman, D.L.: Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. Proceedings of the Royal Society of London. Series B: Biological Sciences **265**(1412) (1998) 2315–2320
21. van Hateren, J.H., van der Schaaf, A.: Independent component filters of natural images compared with simple cells in primary visual cortex. Proceedings: Biological Sciences **265**(1394) (Mar 1998) 359–366
22. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 3642–3649