Learning Transformation Invariance for Object Recognition

Jens Hocke, Thomas Martinetz

Institute for Neuro- and Bioinformatics, University of Lübeck

Abstract. Based on Tomaso Poggio's M-theory, we propose a method to learn transformation invariant representations. Using an artificial dataset, we demonstrate that our supervised method learns invariance to shifts, and on the MNIST data we show first results for learning the unknown transformations underlying handwritten digits.

1 Introduction

Visual object recognition is a challenging task in computer vision. Even small changes to an object's pose can yield dramatic changes to the 2D image in its pixel representation. Therefore, a representation invariant to such changes is mandatory for achieving good recognition rates. Modern approaches to that problem are scale-invariant feature transform (SIFT) [1] for coping with scale invariance and convolutional neural networks [2, 3] for coping with shift invariance.

Recently, the M-theory [4] was proposed explaining how invariance could be implemented in the ventral stream. Besides the theoretical insights on invariance, also a simple algorithm based on this theory is presented to find transformation invariant representations. However, there are two limitations. First, the theory explains only in-plane transformations, and, second, in the algorithm presented, the transformations are assumed to be known in advance. Addressing the later drawback we present a method based on the M-theory to learn invariance to unknown transformations. This enables us to gain (approximate) invariance to complex and unknown transformations.

After introducing the core ideas of the M-theory and describing our approach we demonstrate its potential in an artificial setting and on handwritten digits, assuming these digits undergo complex transformations when written by different people.

2 M-theory

According to the M-theory [4], invariance to a group G of transformations can be achieved in a representation using orbits O. This is the core idea of the M-theory, which we used for our method. In the following we will describe this concept, and refer the reader to [4] for a more exhaustive description of the theory. Here, we use $g \in G$ to denote the group elements, and by $g(\boldsymbol{x})$ we denote the group's action applied to the image $\boldsymbol{x} \in \mathbb{R}^D$. By applying all transformations $g_i \in G$ to some image $\boldsymbol{x}$ an orbit $O_{\boldsymbol{x}} = \{g_i(\boldsymbol{x}) | g_i \in G\}$ is induced. This orbit is unique for the object in $\boldsymbol{x}$, and it is invariant to the transformations in G. For example the group of in-plane rotations would induce an orbit containing all possible rotated versions of the original image $\boldsymbol{x}_1$. The orbit for some other image $\boldsymbol{x}_2 = g_i(\boldsymbol{x}_1)$ that can be obtained from $\boldsymbol{x}_1$ by rotation would be the same, because for both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ all possible rotated versions are contained in the orbit. Of course for some different image $\boldsymbol{x}_3$ that can not be obtained from $\boldsymbol{x}_1$ by rotation the orbit would be different.

For object recognition we would need to generate and compare the orbit of an unknown object to the stored orbit of a known object. It is not clear how to measure the similarity of two obits. One possibility is to use the probability distribution $P_{\boldsymbol{x}}$ induced by the transformations g_i on the image. For these distribution the following holds:

$$\boldsymbol{x}_1 \sim \boldsymbol{x}_2 \iff O_{\boldsymbol{x}_1} = O_{\boldsymbol{x}_2} \iff P_{\boldsymbol{x}_1} = P_{\boldsymbol{x}_2}. \tag{1}$$

However, these probability distributions are extremely high dimensional making it impractical to obtain them. Therefore, we would like to embed the invariance and discrimination properties of the distributions to a space of lower dimension. The Cramér-Wold theorem [5, 4] ensures that these high dimensional probability distributions can be described by D distributions $P_{\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle}$ over one dimensional projections $\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle$, where $\boldsymbol{p}_n, n = 1, \ldots D$ are the projection vectors. To discriminate a finite number of distributions, empirically a small number of projections $N < D$ is sufficient [4].

Instead of transforming the input image $\boldsymbol{x}$, we can also apply the inverse transformation to the projection vectors $\boldsymbol{p}_n$:

$$\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle = \langle \boldsymbol{x}, g_i^{-1}(\boldsymbol{p}_n) \rangle. \tag{2}$$

By applying the transformations to the templates, we avoid transforming every new image. This allows an invariant and discriminative representation in a simple two layer neural network with the transformations stored in the synapses. The first layer generates all the outputs using scalar products of all weight vectors $\boldsymbol{w}_{in} = g_i^{-1}(\boldsymbol{p}_n)$ with the input $\boldsymbol{x}$, and the second layer quantifies the distributions over the outputs of the first layer.

The restriction to groups of transformations allows only few transformations like periodic boundary shifts and in-plane rotations. Other common transformations such as shifts and scaling may not be fully observed by projection vectors of finite length. However, invariance to these partially observable groups can be achieved for a range of parameters and for non-group transformations approximate invariance can be achieved.

3 Invariance Learning

In the original M-theory, the weight vectors $\boldsymbol{w}_{in}$ are derived from the given transformation, e.g., translation or rotation. In our approach we want to learn these weights to be able to adapt to unknown transformations. We quantify the distributions $P_{\langle g_i(\boldsymbol{x}),\boldsymbol{p}_n\rangle}$ by moments m. So every input image $\boldsymbol{x}$ is characterized by

$$y_{nm}(\boldsymbol{x}) = \sum_i^I \left(\boldsymbol{w}_{in}^\top \boldsymbol{x}\right)^m , \tag{3}$$

which is invariant to the transformations $g_i \in G$. In order to obtain a unique and discriminate set of outputs y_{nm}, the number N of projections, the number I of weight vectors per projection and the set of moments need to be set appropriately.

For our supervised approach a set of labeled training images is needed, and there should be multiple images per class available. For every class $c \in C$, moment $m \in M$, and projection $\boldsymbol{p}_n, n = 1, \ldots N$ a target value t_{cmn} is introduced. These target values are used to learn the unknown outputs y_{nm} for every class, with equal outputs for intraclass tuples and different outputs for interclass tuples. The following energy term enforces the moments of the projections to match their target

$$E_{\mathcal{S}} = \sum_k \sum_m \left(t_{cmn} - \sum_i \left(\boldsymbol{w}_{in}^\top \boldsymbol{x}_k\right)^m \right)^2 . \tag{4}$$

By minimizing this term invariance to transformations in the training set is obtained, because the distributions for intraclass tuples are matched. However, this term will not guarantee a discriminative result. Therefore, a second energy term is introduced to enforce a minimum distance between the target vectors of every possible tuple of different classes c and c'

$$E_{\mathcal{D}} = \sum_{c,c'} \max\left(1 - ||\boldsymbol{t}_c - \boldsymbol{t}_{c'}||, 0\right)^2 , \tag{5}$$

with the target vectors $\boldsymbol{t}_c = \left(t_{c,1,1}, t_{c,1,2}, \ldots, t_{|C|,|M|,N}\right)^\top$. The energies $E_{\mathcal{S}}$ and $E_{\mathcal{D}}$ are combined using the weighting factor α

$$E = \alpha E_{\mathcal{S}} + (1-\alpha) E_{\mathcal{D}}. \tag{6}$$

Using this energy term (6) targets t_{cmn} and the weight vectors $\boldsymbol{w}_{in}$ can be learned by gradient optimization, after the targets t_{cmn} and the weight vectors $\boldsymbol{w}_{in}$ have been initialized randomly. In our experience stochastic gradient descent was too slow, and, therefore, we used the Sum of Functions optimizer [6], which in addition to the speed also needs no learning rates to be set.

4 Distance to Center Classification

In case we were able to learn full invariance to a transformation, all images of one class $c*$ will lie exactly on the corresponding target vector $\boldsymbol{t}_{c*}$. If only approximate invariance was achieved, all these images are clustered around $\boldsymbol{t}_{c*}$.

Therefore, the closest target vector determines the class label $c*$ for some image $\boldsymbol{x}$:

$$c* = \arg\min_{c} ||\boldsymbol{y}(\boldsymbol{x}) - \boldsymbol{t}_c||, \tag{7}$$

with $\boldsymbol{y} = \left(y_{,1,1}, y_{1,2}, \ldots, y_{|M|,N}\right)^{\top}$.

5 Experiments

We show first experimental results. Many of the parameters are not optimized, yet. For all the presented results only the second moment was used to quantify the distributions. By setting the weighting parameter α to 0.01, the interclass term was emphasized, which according to our experience leads to faster convergence. The number of projections and the number of weight vectors per projection vary for the experiments, and are described for each experiment separately.

As a proof of concept we used shifted binary patches of size 4×4. 100 patches were generated randomly by setting each pixel to either one or to zero with probability 0.5. Then every patch was shifted using periodic boundary conditions (see Figure 1). On the resulting 1600 training samples, we trained two projections with 16 weight vectors each. We used 16 weight vectors per projection, because we know there are 16 possible shifts. Like the training samples, 160 test samples were obtained from ten random patches by shifting. The orbits of the test samples were then projected with the learned weights using Equation (1). Since we only

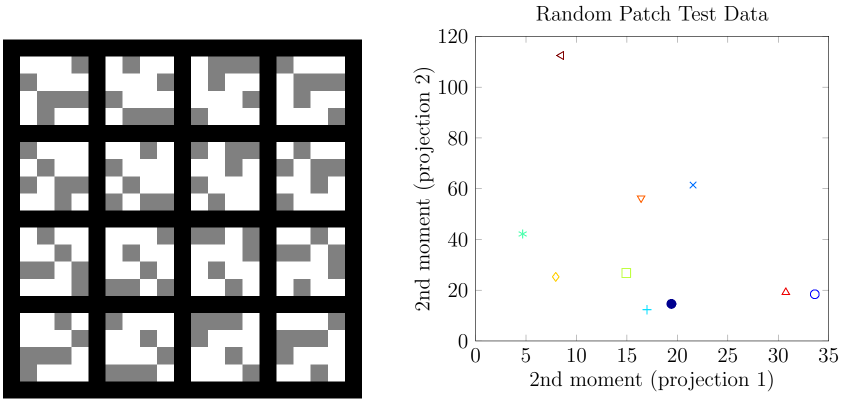

Fig. 1. The left image shows a random example patch in all its 16 possible shifts. The plot on the right shows the two second moments we obtain from projecting the orbits of the test data, i.e., $y_{1,2}$ and $y_{2,2}$ from Equation (1). Each patch is denoted by a different shape and color. All the shifted versions of a patch indeed fall on the same point, demonstrating perfect shift invariance of this representation. The 10 different test patches now can easily be discriminated.

use the second moment, the two projections provide two values for each input image $\boldsymbol{x}$. In Figure 1 we see that the representation is perfectly invariant to the learned transform, because all the shifted versions of a patch fall on a single point.

Going one step further, we tested our method on handwritten digits from the MNIST [3] dataset. It contains 60.000 training and 10.000 test samples. Here, we assume that every sample of a certain digit is a transformed version of a prototype digit. From the training data we learn invariance to the unknown transforms underlying MNIST, which is a much larger challenge than learning the known shifting transform in the experiment above. Since the transforms are unknown, we do not know how to select the number of weight vectors per projection, and in addition the images are of size 28 × 28, therefore many more parameters need to be learned.

For the visualization shown in Figure 2, we trained 2 projections with 20 weight vectors each and again take the second moments. The test data are nicely clustered into the ten digits. Since not all equally labeled digits are perfectly aligned, only an approximately invariant representation was found. However,

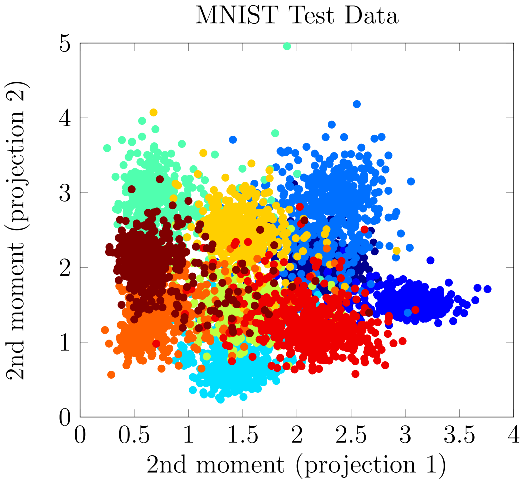

Fig. 2. This plot shows the two second moments we obtain from projecting the orbits of the MNIST test data using our method. Clearly, for every digit the samples form a cluster.

if these two projections we chose for visualization are not enough for perfect separation, we can increase the number of projections. If we use 10 projections, the distance to center classification described in Section 4 achieves 2.86% error rate on the test data significantly improving the 16.63% error rate obtained for the two projection setting. If the distance to center classification is applied in the input pixel space of 768 dimensions, 17.97% of the samples are not classified correctly. This shows how well our method organizes the space.

6 Conclusion

Based on the M-theory, we introduced a supervised learning method to find an invariant representation. In the experiments we showed that our method can learn perfect invariance to periodic boundary shifts. For the much more complex, unknown transformations in MNIST full invariance was not achieved. However, the data was clustered good enough for a decent classification performance. We hope to improve these promising results by a better understanding of the different parameters.

References

1. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2. ICCV '99, Washington, DC, USA, IEEE Computer Society (1999) 1150–1157
2. 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
3. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
4. 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)
5. Cramér, H., Wold, H.: Some theorems on distribution functions. Journal of the London Mathematical Society **s1-11**(4) (1936) 290–294
6. Sohl-Dickstein, J., Poole, B., Ganguli, S.: An adaptive low dimensional quasi-newton sum of functions optimizer. CoRR **abs/1311.2115** (2013)