Learning optimal features for visual pattern recognition

Kai Labusch[a], Udo Siewert[b], Thomas Martinetz[a], and Erhardt Barth[a]

[a]Institute for Neuro- and Bioinformatics, University of Lübeck, Ratzeburger Allee 160, D-23538 Lübeck, Germany;
[b]PLANET intelligent systems GmbH, Residence Park 1-7, D-19065 Raben Steinfeld, Germany

ABSTRACT

The optimal coding hypothesis proposes that the human visual system has adapted to the statistical properties of the environment by the use of relatively simple optimality criteria.

We here (i) discuss how the properties of different models of image coding, i.e. sparseness, decorrelation, and statistical independence are related to each other (ii) propose to evaluate the different models by verifiable performance measures (iii) analyse the classification performance on images of handwritten digits (MNIST data base). We first employ the SPARSENET algorithm (Olshausen, 1998) to derive a local filter basis (on 13×13 pixels windows). We then filter the images in the database (28×28 pixels images of digits) and reduce the dimensionality of the resulting feature space by selecting the locally maximal filter responses. We then train a support vector machine on a training set to classify the digits and report results obtained on a separate test set. Currently, the best state-of-the-art result on the MNIST data base has an error rate of 0,4%. This result, however, has been obtained by using explicit knowledge that is specific to the data (elastic distortion model for digits). We here obtain an error rate of 0,55% which is second best but does not use explicit data specific knowledge. In particular it outperforms by far all methods that do not use data-specific knowledge.

Keywords: natural image statistics, feature extraction, hand-written digit recognition, support vector machine, V1 neurons

1. INTRODUCTION

For a given visual pattern recognition problem such as digit recognition or face finding, one often divides the solution in two parts. Firstly, specific features are extracted from the input data. Secondly, based on the extracted features, a classifier is trained and used to perform the recognition task. Widely-used feature-extraction methods are, for example, the PCA or wavelets.[1]

While the theoretical foundations of machine learning and classifier design are reasonably well understood,[2] the methods for feature extraction are often selected according to heuristic principles that are based on experience and problem-specific knowledge.

Ideally, the feature-extraction method should be adapted to the statistics of input data, i.e., it should extract features that are relevant and useful for the given data set. However, a method that would automatically deliver the optimal features for a given data set is still missing.

In an attempt to make some progress in this direction, we here investigate a method that has been proposed as a model of human vision. The so-called *optimal-coding hypothesis* proposes that the human visual system has adapted to the statistical properties of natural images.[3–6] Different models of image synthesis were proposed, such as the ICA[7,8] and Sparse Coding.[9,10] These models have been successfully employed to mimic properties of V1 cells in the visual cortex.[11–13] Additionally, methods for learning invariances of image transformations such as Slow Feature Analysis[14] were proposed that also reveal strong connections to the properties of the visual system.[15] Assuming that evolution optimised the visual system to cope with a broad range of visual tasks, one would expect that a feature extraction system that is based on the same principles can be used for a wide class of pattern recognition problems as well. We here analyse how well these models perform when used to solve a technical pattern-recognition problem and compare the results with those obtained by state-of-the-art methods.

Recent results show that for natural images the gain in statistical independence obtained by methods like the ICA is rather small, compared to more common methods like the PCA.[16] Nevertheless, we here show that ICA and Sparse Coding can considerably improve recognition performance compared to the PCA in a well-investigated pattern-recognition problem. We choose a benchmark problem of handwritten-digit recognition (Fig 1) for which

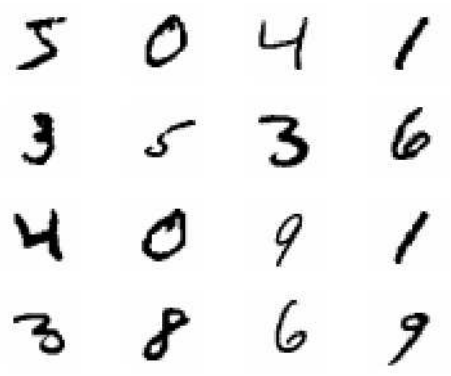

Figure 1. Samples from the MNIST data set of handwritten digits.

many different methods have already been evaluated.[17] In the same framework, we compare results obtained with more common feature-extraction methods such as PCA and Wavelets against ICA and Sparse Coding.

2. FEATURE EXTRACTION

We here describe how we obtain the features to be used for classification. We first describe feature extractors that are basis functions used to encode a local image patch. The feature extractors are either fixed and derived from a specific theoretical framework (in case of the wavelets) or derived from the training data by machine-learning techniques. The feature extractors, or basis functions, are then used to compute the features from the data to be classified.

2.1. Gabor wavelets as feature extractors

The feature extractors of visual area V1 can be modelled as wavelets that do not require any adaptation to the data at hand. Moreover, wavelet-based coding and feature extraction has been shown to be useful for a number of technical applications. As an example of a wavelet function, a Gabor wavelet $\overrightarrow{w}_j$ is determined by its orientation α_j, wavelength λ_j, bandwidth b_j, phase ϕ_j and center $\hat{x}_j$:

$$R = \begin{pmatrix} cos(\alpha_j) & sin(\alpha_j) \\ -sin(\alpha_j) & cos(\alpha j) \end{pmatrix} \tag{1}$$

$$\sigma = \frac{\lambda_j}{\pi}\sqrt{\left(\frac{log(2)}{2}\right)}\frac{2^{b_j}+1}{2^{b_j}-1} \tag{2}$$

$$\overrightarrow{w}_j = e^{-\frac{\|Rx-\hat{x}_j\|}{2*\sigma^2}} cos\left(\frac{2*\pi\,(Rx-\hat{x}_j)}{\lambda_j}+\phi_j\pi\right) \tag{3}$$

Gabor wavelets are a very common method for feature extraction. To assess how a simple set of Gabor filters compares with the adaptive methods described below, we included them in our experiments.

2.2. Unsupervised learning of feature extractors

In the following we consider different unsupervised learning methods which can be used to compute basis functions that are representative of the data in the sense that the input data can be reconstructed from the basis functions. The learning methods differ with respect to the target criterion that defines which basis functions are selected and they either postulate perfect reconstruction or allow for reconstruction errors (noise) in the model.

Furthermore, the features that we would like to obtain should be local, i.e., encode the local properties of an image. Therefore we consider image patches $I(x, y)$ of size $N \times N$ that are randomly drawn from the full-size

input images I_F. We treat the $I(x, y)$ as column vectors that contain the pixel values of the image patch arranged in an appropriate scheme.

We aim at learning basis functions $\overrightarrow{w}_j$ of the image patches $I(x, y)$ that allow their reconstruction only from the $\overrightarrow{w}_j$ and some coefficients x_j. We use PCA, ICA, and Sparse Coding to learn such features. All the above methods can be described in the same image patch generation framework as follows. An image patch $I(x, y)$ is obtained from a linear combination $\overrightarrow{x}$ of the features $\overrightarrow{w}_j$. In addition, we may allow for an additive error term $\overrightarrow{\epsilon}$ that corresponds to a certain amount of noise that is present in the image patch generation model.

$$I(x,y) = \sum_{j=1}^{M} \overrightarrow{w}_j x_j + \overrightarrow{\epsilon} \tag{4}$$

$$= W\overrightarrow{x} + \overrightarrow{\epsilon} \tag{5}$$

2.2.1. PCA

Equation (5) can be seen as basic image patch generation model of the PCA by assuming that $\overrightarrow{\epsilon} = 0$. In this model, the x_j are pairwise uncorrelated. The $\overrightarrow{w}_j$ form an orthogonal basis of the $I(x, y)$. If a sufficient amount of observed data $I(x, y)$ is available, the $\overrightarrow{w}$ can be obtained as the eigenvectors of the covariance matrix $C = \langle I(x, y)I(x, y)^T \rangle$ of the observed distribution of the $I(x, y)$. PCA can be motivated by other considerations such as finding an orthogonal basis of the $I(x, y)$ that consists of the directions of maximum variance of the pattern data or compression with minimal mean square error.

2.2.2. ICA

Equation (5) also can be seen as the image patch generation model of noise-free ICA. Again $\overrightarrow{\epsilon} = 0$ is assumed. Furthermore, the ICA postulates that the x_j are pairwise statistically independent and stem from non-Gaussian distributions. The ICA can be interpreted as a generalisation of the PCA that, in contrast to the PCA, not only considers second-order statistics but also higher-order statistical properties of the pattern data such as the kurtosis. The optimisation goal of ICA is not only the decorrelation of the coefficients x_j but their statistical independence. We used an ICA approach[18] that maximises the non-Gaussianity of the x_j[7] to achieve this goal. This can result in coefficients x_j that are sparse. Other motivations for the ICA exist, for example based on the information maximisation principle.[8]

2.2.3. Sparse Coding

Within the Sparse Coding approach, equation (5) postulates an image-patch generation model where the x_j stem from sparse (leptocurtic) distributions. Hence, the primary goal of sparse coding is the maximisation of the sparsity of the coefficients x_j. The reconstruction error (noise) is assumed to be Gaussian. The model now allows for balancing the reconstruction error $\overrightarrow{\epsilon}$ against the sparsity of the coefficients x_j. The maximisation of non-gaussianity that can be obtained by standard ICA and the maximisation of the sparsity may lead to similar results depending on the data. Therefore Sparse Coding can be seen as a more realistic generalisation of the PCA (compared to the ICA) since it allows for a certain level of noise in the signal. There are different sparse-coding approaches available.[9,19] We use the well-known Sparsenet algorithm[9] that allows for controlling the reconstruction error $\overrightarrow{\epsilon}$ by the weight of the explicit sparseness criterion in the target function that is optimised.

2.3. Obtaining features from basis functions

We now use the feature extractors (basis functions) $\overrightarrow{w}_j$ described above by assuming that the extracted features represent relevant aspects of the data that are useful for the pattern recognition problem to be solved.

To extract the features, we measure the similarity between each possible $N \times N$ patch $I(x, y)$ of a given input image and each feature $\overrightarrow{w}_j$ by

$$D_j(x,y) = \frac{\langle I(x,y), \overrightarrow{w}_j \rangle}{\|I(x,y)\| \|\overrightarrow{w}_j\|}. \tag{6}$$

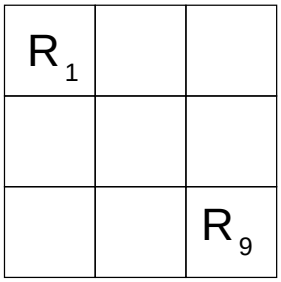

Figure 2. The image is divided into a set of regular non-overlapping regions. In each of the image regions R_i we take the maximum similarity value with respect to each basis function as defined in equation (7).

This is equivalent to a normalised convolution of the pattern images with the basis functions $\overrightarrow{w}_j$. Furthermore, we assume that the relevant features cannot be localised at a certain pixel position, but that they are typically located in a restricted region of the pattern image, i.e., we allow for some spatial uncertainty. Therefore, we divide the input image into a set of regular, non-overlapping regions $R_i, i = 1, ..., M^2$ and take as local feature the maximum similarity of each region with respect to each basis function $\overrightarrow{w}_j$ (see figure 2):

$$D_j(R_i) = \max_{x,y \in R_i} D_j(x, y) \tag{7}$$

The final feature vector (that is finally fed to the classifier) of each input image consists of the maximum similarity values of all regions with respect to all basis functions:

$$f_I = \left(D_j(R_i)\right), i = 1, ..., M^2, j = 1, ..., N^2 \ . \tag{8}$$

3. SVM CLASSIFIER

The obtained feature vectors are used to train a 2-norm soft margin SVM with Gaussian kernels with the SoftDoubleMaxMinOver learning algorithm.[20] On the MNIST set, we need to solve a ten-class problem since we have to differentiate ten digits. To accomplish this task using a SVM we trained 45 two-class classifiers that each separate two different digits (one against one). The decisions of all the two-class classifiers are then counted and finally the class with the majority of votes is selected. The hyperparameters of the SVM (kernel width γ and softness parameter C) were optimised by cross-validation.

4. EXPERIMENTS

All the described methods for unsupervised learning of feature extractors were tested in the same setting in order to ensure comparability. As mentioned before, the MNIST set of handwritten digits was used as a benchmark because the data set is quite popular.[21–23] The MNIST data set consists of 60000 training and 10000 test images of size 28×28. Currently, the best results reported on the MNIST data set were obtained with convolutional neural networks and elastic distortions (0.4% error rate[23]) and Virtual SVM with deskewing and jittering preprocessing (0.56% error rate[22]). The best result obtained so far with methods that do not use additional data-specific knowledge is 0.95% for a LeNet-5 network.[21,24] The procedure described in the following was applied in the same way for each feature-extraction method mentioned in section 2.2.

The feature extractors $\overrightarrow{w}_j$ were trained on image patches $I(x, y)$ of size 13×13 that were extracted at random positions from a subset of the training set. As mentioned before we set the number of feature extractors equal to the number of dimensions of the image patches, resulting in 169 features $\overrightarrow{w}_j$ (figure 3,4, 5).

Note that the preprocessing described in Fig 5 has been applied only in the learning phase to determine the optimal basis functions, but not for the actual feature extraction for actual classfication.

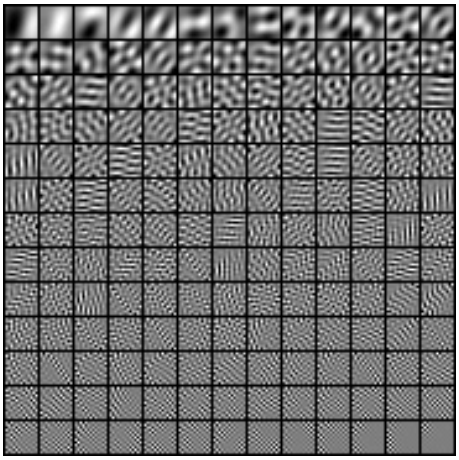

Figure 3. Basis functions obtained for randomly drawn image patches of size 13×13 by using the PCA.

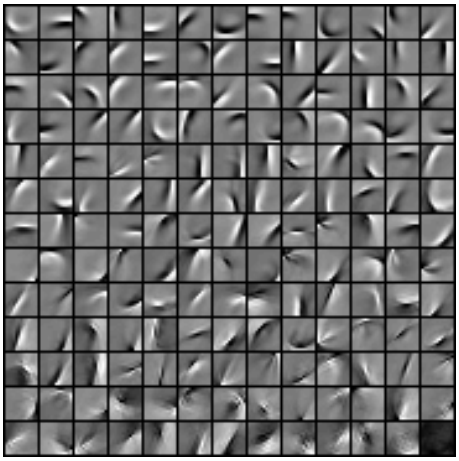

Figure 4. ICA basis functions of randomly drawn image patches of size 13×13 obtained with the FastICA algorithm by using its default parameters.

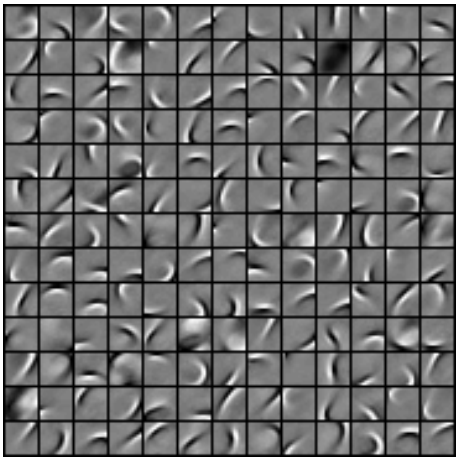

Figure 5. Basis functions of randomly drawn image patches of size 13×13 obtained from the Sparsenet algorithm. In this case we applied a bandpass filter to the input images (before extracting the image patches) as proposed in [9]. Additionally we set the mean pixel variance to 1 in order to achieve convergence of the algorithm. We performed 10000 training iterations using a batch size of 500 samples. The noise variance parameter of the algorithm was set to 0.086 while the β parameter was set to 1.0.

We obtained feature extractors $\overrightarrow{w}_j$ that were used as described in section 2.3 to extract a feature vector f_I for each of the 60000 training samples. We did not apply any preprocessing to the images before the feature extraction step. We used 9 maximum regions R_i of size 9×9. As a consequence, we did not consider the bottom row and the last column of the convolution result in (6) for feature extraction. Due to the feature extraction procedure, the dimensionality of the training data of the classifier increased from $28 \times 28 = 784$ (in case of the raw data) to $9 * 169 = 1521$.

A set of 160 Gabor filters (figure 6) was used with the same feature-extraction procedure to obtain training data of dimensionality $9 * 160 = 1440$.

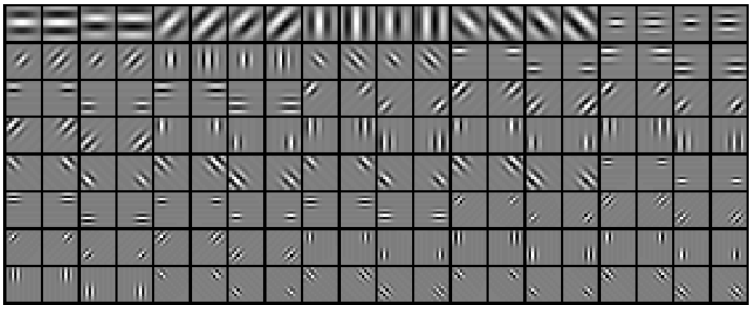

Figure 6. The Gabor filters that were used are shown for comparison.

The hyperparameters of the SVM were optimised by 7-fold cross validation using only training features. We used the same hyperparameters for all the different two-class classifiers. Each of the seven realisations of test and training data used for cross validation consisted of disjoint sets of 10000 samples. We took the hyperparameters providing the best mean classification error on the cross validation test sets in a grid search over γ and C. Using the best hyperparameters the final classifier was trained on the entire set of feature vectors of all 60000 training samples.

Finally, the feature extraction was applied to the input images of the separate test set consisting of 10000 samples. The SVM was then tested on these feature vectors.

method	#SVs per digit										#SVs	error rate
	0	1	2	3	4	5	6	7	8	9		
raw data	1235	622	2149	1891	1681	2071	1365	1492	2342	2014	16862	0.0142
PCA	684	346	1087	965	884	961	835	759	1249	1147	8917	0.0092
Gabor wavelets	610	315	1014	861	760	843	768	755	1207	888	8021	0.0075
ICA	338	379	556	548	547	524	419	613	737	759	5420	0.0058
Sparsenet	403	392	648	593	529	580	473	682	816	747	5863	0.0055

Table 1. The table shows the number of SVs from each digit class as well as the overall number of SVs together with the error rate obtained on the test set consisting of 10000 samples.

5. RESULTS

Table 1 shows the number of support vectors for the 10 different classes, the overall number of support vectors as well as the error rate on the MNIST test set.

All methods significantly outperform the direct classification of the raw data, where no features have been extracted. Results obtained for Gabor wavelets, ICA, and Sparse Coding are significantly better than those obtained for the PCA. Note that ICA and Sparse Coding provide similar results on this data set. Overall, the improvement of the test error is associated with a decrease in the number of support vectors.

The virtual SVM result of 0.56% errors reported in [22] is very close to our best result, but has the disadvantage of requiring a very high number of support vectors whereas our best result uses a much smaller number of support vectors. This indicates that the feature extraction we perform successfully implements invariances of the problem of handwritten digit recognition.

Furthermore, our experiments show that the very good results obtained with Sparsenet and ICA are stable for small changes of feature extractor size and size of maximum region (results not shown).

6. CONCLUSION

We proposed a feature extraction method based on unsupervised learning algorithms and we showed that it outperforms all state-of-the-art methods on the MNIST data set except for one. The one which is still the best uses explicit knowledge about digits, which we do not. In particular, our method provides by far the best result of all methods that do not use prior knowledge that is specific to the handwritten digit recognition problem (a comprehensive list of results can be found on the internet*).

Our method is quite simple and may be applied to a broad range of visual pattern recognition problems. The free parameters of the method such as the size and location of the regions R_i, or the size and number of the features $\overrightarrow{w}_j$ may be optimised with respect to the recognition problem at hand.

In our experiments the features obtained from the Sparsenet algorithm provide the best results. This demonstrates how knowledge about the organisation of the visual system can be applied succesfully to a technical problem.

The noise level parameter of the Sparsenet algorithm was chosen such as to guarantee stable convergence. We believe that by optimisation of the sparseness parameter further improvements could be obtained.

7. ACKNOWLEDGEMENT

We thank the PLANET intelligent systems GmbH for supporting this research.

*http://yann.lecun.com/exdb/mnist/

REFERENCES

1. J. G. Daugman, "Complete discrete 2-D gabor transforms by neural networks for image analysis and compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing* **36**(7), pp. 1169–1179, 1988.
2. V. N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
3. C. Zetzsche, E. Barth, and B. Wegmann, "The importance of intrinsically two-dimensional image features in biological vision and picture coding," in *Digital Images and Human Vision*, A. B. Watson, ed., pp. 109–38, MIT Press, Oct. 1993.
4. D. J. Field, "What is the goal of sensory coding?," *Neural Computation* **6**(4), pp. 559–601, 1994.
5. B. A. Olshausen and D. J. Field, "Natural image statistics and efficient coding," *Network* (7), pp. 333–339, 1996.
6. E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual Review of Neuroscience* **24**, pp. 1193–1216, 2001.
7. A. Hyvärinen, "Survey on independent component analysis," *Neural Computing Surveys* **2**, pp. 94–128, 1999.
8. A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation* **7**(6), pp. 1129–1159, 1995.
9. B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature* (381), pp. 607–609, 1996.
10. M. S. Lewicki and T. J. Sejnowski, "Learning nonlinear overcomplete representations for efficient coding," in *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pp. 556–562, MIT Press, (Cambridge, MA, USA), 1998.
11. A. J. Bell and T. J. Sejnowski, "The "independent components" of natural scenes are edge filters.," *Vision Res* **37**, pp. 3327–3338, December 1997.
12. B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?," *Vision Research* **37**(23), pp. 3311–3325, 1997.
13. B. Olshausen and D. Field, "Sparse coding of natural images produces localized, oriented, bandpass receptive fields.," 1995.
14. L. Wiskott and T. J. Sejnowski, "Slow feature analysis: unsupervised learning of invariances," *Neural Computation* **14**(4), pp. 715–770, 2002.
15. P. Berkes and L. Wiskott, "Applying Slow Feature Analysis to Image Sequences Yields a Rich Repertoire of Complex Cell Properties," in *ICANN '02: Proceedings of the International Conference on Artificial Neural Networks*, pp. 81–86, Springer-Verlag, (London, UK), 2002.
16. M. Bethge, "Factorial coding of natural images: how effective are linear models in removing higher-order dependencies?," *J Opt Soc Am A Opt Image Sci Vis* **23**, pp. 1253–1268, June 2006.
17. Y. LeCun, "MNIST handwritten digit database, NEC research institute."
18. A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks* **10**(3), pp. 626–634, 1999.
19. M. S. Lewicki and T. J. Sejnowski, "Learning Overcomplete Representations.," *Neural Computation* **12**(2), pp. 337–365, 2000.
20. T. Martinetz, K. Labusch, and D. Schneegaß, "SoftDoubleMinOver: A Simple Procedure for Maximum Margin Classification.," in *Artificial Neural Networks: Biological Inspirations. ICANN 2005: 15th International Conference. Proceedings, Part II*, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, eds., pp. 301–306, 2005.
21. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE* **86**, pp. 2278–2324, November 1998.
22. D. Decoste and B. Schölkopf, "Training Invariant Support Vector Machines," *Mach. Learn.* **46**(1-3), pp. 161–190, 2002.
23. P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," in *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, p. 958, IEEE Computer Society, (Washington, DC, USA), 2003.
24. Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," pp. 255–258, 1998.