ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Interpreting Hybrid AI through Autodecoded Latent Space Entities

Roland J. Veen[1,2] and Christodoulos Hadjichristodoulou[1] and Michael Biehl[1]

1- Univ. of Groningen, Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Groningen, The Netherlands

2- Medical Research Council Laboratory of Medical Sciences (MRC LMS) London, United Kingdom

Abstract. Explainable AI models and methods have seen a rise in interest in recent years as a reaction to the widespread use of neural networks and similar black-box models in machine learning. In this project, we combine explainable, prototype-based systems and neural networks in an effort to benefit from both approaches. Specifically, we employ Generalized Matrix Relevance Learning Vector Quantization in combination with autoencoder networks. This allows us to perform automated non-linear feature extraction from high-dimensional inputs before feeding them into LVQ for classification. Moreover, the approach enables the mapping of the low-dimensional representatives and relevances back to the original feature space for visual inspection and interpretation.

1 Introduction

In recent decades, there have been many rapid developments in the area of neural networks [1]. They show state-of-the-art performance in various applications, including classification, segmentation, and detection. However, a major drawback is that they often are *black-box* systems. Frequently, their inner workings cannot be interpreted by humans easily, and it is in general difficult to obtain insight into what affects the actual decisions of a given system. Thus, there is an increasing interest in explainable models and algorithms [2]. In some cases, the ability to properly explain why exactly an intelligent system behaved the way it did is more important than how accurate or fast it is.

Learning Vector Quantization (LVQ) [3] constitutes a family of intuitive, understandable algorithms. It addresses classification tasks and offers interpretability by identifying prototypes. Because prototypes are defined in the same space as the data presented, they can be interpreted and compared directly with individual data points. One or several prototypes represent each class, and a given input is classified according to its distances from all prototypes.

Applying LVQ to very high-dimensional data, e.g., images, can be challenging and often requires dimensional reduction as a pre-processing step. In previous work, convolutional neural networks (CNNs) [4] or the encoder of an autoencoder (AE) [5] have been combined with LVQ with great success, but with the drawback of losing a degree of interpretability.

Here, we propose to combine a variant of LVQ, Generalized Matrix Relevance

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

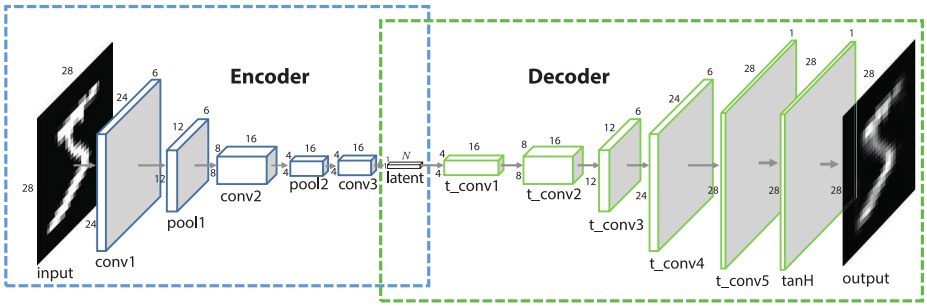


Fig. 1: Architecture of the autoencoder, redrawn after [11] and adapted.

LVQ (GMLVQ[1]) [6], with custom convolutional auto-encoders (CAE). The GMLVQ classifier is trained and operates in the latent space. The CAE allows the prototypes to be mapped back to the original high-dim. feature space.

Previous work with similar independent ideas [7] combines a Neural Network with an AE and a custom prototype layer. However, that solution is less transparent than our method using GMLVQ.

2 Methods

All experiments were carried out using MATLAB r2023a [8] and its *Deep Learning* toolbox. For GMLVQ training, we used the *No-nonsense beginner's GMLVQ toolbox* [9] with default parameters apart from the number of total steps. We performed part of the experiments on the *Hábrók* HPC cluster of the University of Groningen. The MNIST dataset [10] of handwritten digits serves as a testbed to evaluate and illustrate the suggested method.

Autoencoder architecture and training: We built an autoencoder with convolutional layers based on the architecture proposed in [11]. Because the *Deep Learning* toolbox does not support unpooling layers, we replaced them with transposed convolution layers that serve the same purpose. We also added a hyperbolic tangent layer at the end of the stack. The final architecture is presented in Fig. 1. For training, we use the ADAM optimizer [12] with an initial learning rate of 10^{-4} and otherwise default parameters. The pixel-wise Mean Squared Error (MSE) between the reconstructed image and the original is used as the loss function.

Fusion of GMLVQ and CAE: The core idea is combining GMLVQ with CAE to classify a high-dimensional dataset and preserve interpretability simultaneously. Specifically, we train the CAE described above on the MNIST dataset with $28 \times 28 = 784$ input dimensions. Then we isolate the encoder part of the network. We use the activations in the latent space as low-level representations of the input images and train the GMLVQ algorithm in this low-dimensional

[1] GMLVQ is more powerful and versatile than vanilla LVQ due to a generalized cost function and a feature scaling relevance matrix, see [6].

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

space. After training is complete, we obtain prototypes and pass them through the decoder, which allows us to project them back to the original feature space and display them as images again. The process of classifying a new image involves encoding it using the encoder part of the autoencoder, finding the closest prototype based on the metric defined by the relevance matrix, and assigning the corresponding label to the image.

One of the benefits of the scheme is the lower computational cost for the calculation of distances between datapoints and prototypes due to the reduced dimensionality. By design, CAE aim to capture the most salient features while eliminating unimportant ones that could also harm the performance of the classifier. Another key aspect of this method is interpretability of the prototypes obtained. As mentioned above, we can use the decoder to map the prototypes back to the original feature space, display them as images, and visually inspect how well they represent their classes.

The decoder can also be used to obtain information about the relevance matrix. The relevance matrix is expressed as $\Lambda = \sum_{j=1}^{N} \lambda_j \mathbf{u}_j \mathbf{u}_j^\top$ with non-negative eigenvalues λ_j and eigenvectors $\mathbf{u}_j$ whose dimensionality matches the latent space of the CAE. In two-class problems it is typically dominated by the leading eigenvector with $\lambda_1 \approx 1$, see e.g., [13]. Hence, we only project the principal eigenvector back to the image space for visual inspection.

Experimental Configurations: The autoencoder was trained for 20-30 epochs for all experiments. The latent size N varied between experiments, chosen empirically by selecting the smallest size that resulted in an acceptable CAE reproduction loss. The GMLVQ classifier was trained using the default hyperparameters defined in the toolbox [9] for 30 batch gradient steps. For each experiment, we trained the system a total of 10 times and used the average metrics for validation purposes. We did not attempt to fine-tune the hyperparameters of the algorithm since it was not the focus of this proof-of-concept study.

Performance Criteria: We evaluated the suggested method both quantitatively and qualitatively. Specifically for binary classification problems, we used the AUROC [14] for evaluation. In addition, we evaluated the method on the basis of the interpretability of the decoded prototypes.

3 Results and Discussion

In this section, we present the experiments performed mainly in an exploratory manner, as well as the corresponding results. We also discuss our findings and their significance.

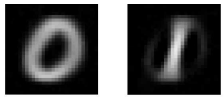
One digit per class: We trained the classifier to separate (a) handwritten *ones* from *zeros* and (b) *twos* from *threes*. Here, the encoder reduced the number of features to $N = 5$. We report the accuracies and AUROC in Table 1 and present decoded representations of the prototypes obtained in Figure 2.

We observe that the algorithm yields very plausible representations of the classes in image space. It also achieved very good classification performance as expected in this relatively simple and limited experiment.

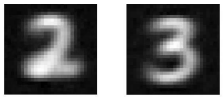
ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Classification	Accuracy	AUROC
0/1	0.995	0.999
2/3	0.949	0.988

Table 1: Accuracy and AUROC for binary tasks, denoted in the form i/j, where i and j are the two digits between which the classifier has to distinguish.



(a) 0/1 classification



(b) 2/3 classification

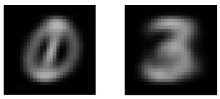
Fig. 2: Two-class problems with one digit per class. Decoded prototypes are shown as images.

Multiple digits per class: As examples of slightly more complex classification tasks we considered the grouping of multiple digits into classes. In a first setting, we grouped digits *zero* and *one* in a single class while the second comprises *twos* and *threes*. In the following we refer to this problem as $\{0,1\}/\{2,3\}$. Next we considered classes comprising visually similar pairs of digits, for instance the discrimination of $\{3,8\}/\{1,7\}$.

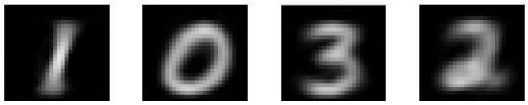
In both cases we performed experiments with one prototype and, alternatively, two prototypes per class. Here, the bottleneck dimension was $N = 10$. Table 2 displays the achieved accuracies and AUROC, while in Fig. 3 we present the obtained decoded prototypes. The performance of the classifier does not change significantly upon increasing the number of prototypes, but it appears to be affected by the specific choice of digits. Due to the higher in-class similarity, groups $\{3,8\}$ and $\{1,7\}$ were separated slightly better than $\{0,1\}$ and $\{2,3\}$.

Obviously, the outcome in terms of the decoded prototypes is strongly affected by allowing more representatives per class. As shown in Fig. 3, single prototypes resemble a superposition of the grouped digits. Systems with two prototypes per class do resolve and represent the individual digits, although the corresponding label information was not available in the training.

Prototype difference and the principal eigenvector: Intuitively, the eigenvec-



(a) One prototype per class



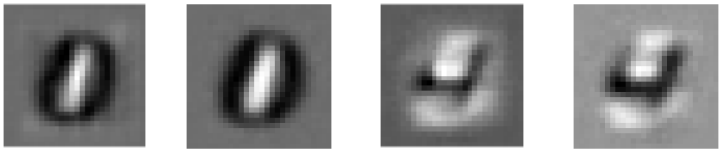
(b) Two prototypes per class

Fig. 3: Two-class problems with two digits per class. Decoded prototypes for grouped classes $\{0,1\}$ and $\{2,3\}$ are shown and interpreted as images.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

task	prototypes	Acc.	AUROC	prototypes	Acc.	AUROC
{0,1} / {2,3}	1 per class	0.915	0.972	2 per class	0.929	0.982
{1,7} / {3,8}	1 per class	0.940	0.976	2 per class	0.939	0.989

Table 2: Accuracies (Acc.) and AUROC for grouped classes



(a) (b) (c) (d)

Fig. 4: Visualization of the decoded eigenvectors (a and c) and the difference between the two prototypes (b and d) for the experiments 0/1 and 4/5.

tors of the relevance matrix in GMLVQ should capture the directions in which the classes differ the most. We compared the principal eigenvector with the pixel-wise difference between the two prototypes of the binary classification tasks. Specifically, we decoded the principal eigenvector of the relevance matrix and interpreted it as an image. Results were inspected visually and in terms of the cosine similarity. We report our findings in Table 3 and in Figure 4 where we show some examples for illustration purposes. We see high similarity scores for most cases, as one would generally expect. However, in a couple of tasks, the measure is significantly lower; see the experiment with digits *four* and *five*. This effect may correlate with the lower accuracy obtained for these tasks, which will be investigated more thoroughly.

Experiment	0/1	2/3	4/5	6/7	{0,1}/{2,3}	{1,7}/{3,8}
Cos. Sim.	0.965	0.864	0.539	0.792	0.943	0.807

Table 3: Cosine similarity between decoded eigenvector and prototype difference

4 Conclusion and Future Work

In this project we showcased our proposed model of combining GMLVQ with CAE as a way to enable automatic feature extraction from high-dimensional data, while at the same time creating a pathway in the opposite direction, from the low-dimensional representation back to the original feature space. With these tools, we are able to interpret the prototypes that were trained in the latent space as images again, giving this approach the benefit of explainability.

We do not attempt to compete with the state-of-the-art models in the space of classification, but based on the results, we make the case that the method shows promise and, with further fine-tuning, could be used to a satisfactory

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

degree if properties like explainability and prototype-based learning are desired. Aside from performance evaluation, these experiments allowed us to examine some properties of the resulting models, most notably the visualization of the eigenvectors of the relevance matrix.

Future Work: Forthcoming studies will address the benchmarking of the method on various established datasets for classification and comparison with other existing models. This will require the fine-tuning of a relatively large number of hyperparameters, but it should provide better insights into the strengths of the method and its potential for real-world applications. Another topic that deserves further attention is the relationship between distances in the encoded space and after decoding.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `https://www.deeplearningbook.org`.

[2] A. Bibal and B. Frénay. Interpretability of machine learning models and representations: an introduction. In M. Verleysen, editor, *Proc. Europ. Symp. Artificial Neural Networks (ESANN)*, pages 77–82, 2016.

[3] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995.

[4] T. Villmann, M. Biehl, A. Villmann, and S. Sarajalew. Fusion of deep learning architectures, multilayer feedforward networks and learning vector quantizers for deep classification learning. In *Proc. 12th Intl. Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)*. IEEE, 2017. 6 pages.

[5] C.F.F. Costa-Filho, J.V. Negreiro, and M.G.F. Costa. Multimodal Biometric System Based on Autoencoders and Learning Vector Quantization. In T.F. Bastos-Filho, E.M. de Oliveira Caldeira, and A. Frizera-Neto, editors, *XXVII Brazilian Congress on Biomedical Engineering*, pages 1611–1617, Cham, 2022. Springer International Publishing.

[6] P. Schneider, M. Biehl, and B. Hammer. Adaptive Relevance Matrices in Learning Vector Quantization. *Neural Computation*, 21:3532–3561, 2009.

[7] O. Li, H. Liu, C. Chen, and C. Rudin. Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions. In *Proc. of the 32nd AAAI Conf. and 30th IAAI Conf. and 8th EAAI Symp.* AAAI Press, 2018.

[8] The MathWorks Inc. Matlab version: 9.14.0 (r2023a), 2023. `https://www.mathworks.com`.

[9] M. Biehl, R.J. Veen, and F. Westerman. A no-nonsense beginner's tool for GMLVQ, Sep 2021. `https://www.cs.rug.nl/~biehl/gmlvq`.

[10] Y. LeCun, C. Cortes, and C. Burges. MNIST handwritten digit database. `https://yann.lecun.com/exdb/mnist/`.

[11] F. Li, H. Qiao, B. Zhang, and X. Xi. Discriminatively boosted image clustering with fully convolutional auto-encoders. *CoRR*, abs/1703.07980, 2017.

[12] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] M. Biehl, B. Hammer, F.-M. Schleif, P. Schneider, and T. Villmann. Stationarity of Matrix Relevance LVQ. In *Proc. IEEE International Joint Conference on Neural Networks (IJCNN 2015)*. IEEE, 2016.

[14] T. Fawcett. Introduction to ROC analysis. *Pattern Recognit. Lett.*, 27:861–874, 06 2006.