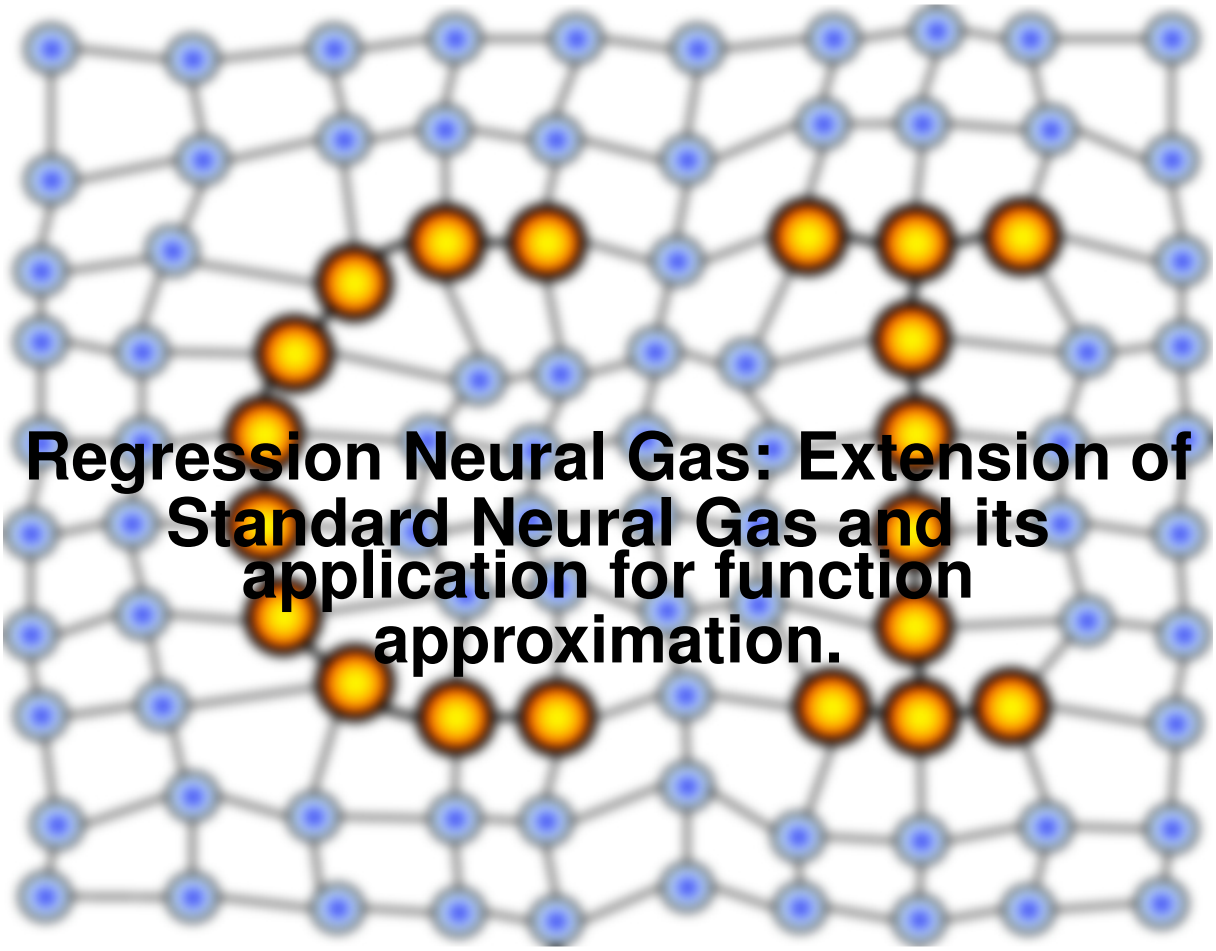
MACHINE LEARNING REPORTS



Regression Neural Gas: Extension of Standard Neural Gas and its application for function approximation.

Report 01/2023
Submitted: 25.03.2023
Published:28.03.2023

Ronny Schubert[1], Marika Kaden [1] and Thomas Villmann[1]
(1) University of Applied Sciences Mittweida
Technikumplatz 17, 09648 Mittweida - Germany

Machine Learning Reports
http://www.techfak.uni-bielefeld.de/∼fschleif/mlr/mlr.html

Abstract

In this work we propose extensions of the Neural Gas algorithm and its application for time-series prediction, i.e. more general for function approximation. We consider mathematical aspects of respective prototype-based frameworks for regression learning and provide generalizations of the fuzzy-labeled Neural Gas for this setting.

1 Introduction

Learning of complex relations between data is still a challenging task, if the respective models is required to be interpretable to a great extent. For those task, prototype-based models for underlying data space partition by vector quantization seems to be promising and robust methods, which are recommended for use in complex learning systems to maintain robsutness, stability and interpretability [*Li et al. (2018)*, *Rudin et al. (2022)* *Lisboa et al. (2023)*].

Various methods have been proposed to combine data (vector) representation algorithms like k-means or *self-organizing maps* (**SOM**) with approximation or regression techniques, see for example *Moody and Darken (1989)* and *Hecht-Nielsen (1987)*, respectively. In particular, the *Counterpropagation* network (**CPN**) proposed by *Hecht-Nielsen (1987)* consists of a SOM-layer followed by a perceptron layer for classification or regression predictions. However, the training of the SOM-layer is independent of the subsequent perceptron-layer. Hence, the unsupervised representation learning is not adjusted to the following supervised learning for the prediction task. In contrast, the *radial basis function* network (**RBFN**) by *Moody and Darken (1989)* can be trained in both modes hybrid or supervised. In this paper, the hybrid mode describes those algorithms where the representation algorithm is only used for the partitioning of the data space independently from the subsequent or simultaneously learned regression task. The supervised mode algorithms aim to adjust the representation learning alsi in dependence of the regression (or classification) problem. In this sense, a supervised extension of the CPN was proposed by *Kaden et al. (2021)* and a generalization to RBFN can be found in *Poggio and Girosi (1989)*. Furthermore, *Grbovic and Vucetic (2009)* extended the *learning vector quantization* algorithm established by *Kohonen (1995)* for a classification learning to a regression approach. In this work, we suggest to apply *Neural Gas* network (**NG**) proposed by *Martinetz et al. (1993)* for robust ancd accurate representation learning, which has been used also for time-series prediction. Yet, we extend this approach allowing on the one hand side polynomial transformations of the data and, on the other hand, incorporate neighborhood information of the representing prototypes also into the regression learning scheme.
In section 2 - 3 we provide the necessary mathematical theory and basis for our work as well as present NG and other related approaches for regression learning. In section 4 we present our new model and 5 concludes this technical report by final remarks.

2 The Neural Gas Network

The Neural Gas Network (NG) is an unsupervised vector quantization method improving k-means and self-organizing maps [*Kohonen (1982)*, *Martinetz et al. (1993)*]. The network consists of artificial neurons $\mathcal{N} = \{1, ..., k\} \subset \mathbb{N}^+$ equipped with weight vectors but here interpreted as prototypes (reference vectors) $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_k\} \subset \mathbb{R}^n$ for data representation. To create a network response, we suppose a given data sample (stimulus) $\boldsymbol{x}$ from the data set $\boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^n$. The response of the network to the given stimulus $\boldsymbol{x}$ obeys the *winner-takes-all* rule (WTA-rule):

$$\nu(\boldsymbol{x}, \boldsymbol{\mathcal{P}}) = \underset{j \in \mathcal{N}}{\arg\min}\, d(\boldsymbol{x}, \boldsymbol{p}_j)\,, \tag{1}$$

with $d(\boldsymbol{x}, \boldsymbol{p})$ as a differentiable dissimilarity measure, which commonly is chosen as the squared Euclidean norm, i.e. $d(\boldsymbol{x}, \boldsymbol{p}) = ||\boldsymbol{x} - \boldsymbol{p}||_E^2 = (\boldsymbol{x} - \boldsymbol{p})^2$. Thus, the stimulus $\boldsymbol{x}$ gets assigned to the neuron $s = \nu(\boldsymbol{x}, \boldsymbol{\mathcal{P}}) \in \mathcal{N}$ also denoted as winner neuron, which results in the smallest distance $d(\boldsymbol{x}, \boldsymbol{p}_s)$ of $\boldsymbol{x}$ to the prototype-set $\boldsymbol{\mathcal{P}}$. Further, the set $\boldsymbol{\mathcal{P}}$ generates a Voronoï decomposition $\mathcal{V} = \{V_1, ..., V_k\}$ of the data space $\boldsymbol{\mathcal{X}}$ by means of the WTA-rule (1), i.e.

$$V_j = \{\boldsymbol{x} \in \boldsymbol{\mathcal{X}} | \nu(\boldsymbol{x}, \boldsymbol{\mathcal{P}}) = j\}$$

is valid.

To adapt NG to the dataset $\boldsymbol{\mathcal{X}}$, *Martinetz et al. (1993)* proposed an adaptation rule for the prototypes $\boldsymbol{\mathcal{P}}$ which incorporates a kind of neighbourhood cooperativeness of the neurons in implicit dependence on the winning neuron s by means of their assigned prototypes: Particularly, for a stimulus $\boldsymbol{x}$ every prototype p_j is updated according to

$$\Delta p_j = -\epsilon \cdot h_\lambda(\boldsymbol{x}, \boldsymbol{p}_j, \boldsymbol{\mathcal{P}}) \cdot \frac{\partial d(\boldsymbol{x}, \boldsymbol{p}_j)}{\partial \boldsymbol{p}_j} . \tag{2}$$

where the function $h_\lambda(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\mathcal{P}})$ realizes the *neighbourhood-cooperativeness* and is defined as

$$h_\lambda(\boldsymbol{x}, \boldsymbol{p_j}, \boldsymbol{\mathcal{P}}) = exp\left(-\frac{rk(\boldsymbol{x}, \boldsymbol{p_j}, \boldsymbol{\mathcal{P}})}{\lambda}\right) , \tag{3}$$

which depends on the current winning rank $rk(\boldsymbol{x}, \boldsymbol{p_j}, \boldsymbol{\mathcal{P}})$ of the prototype $\boldsymbol{p_j}$ for a given stimulus $\boldsymbol{x}$ and, hence, implicitly depending on the winning prototype p_s. Thus, all prototypes $\boldsymbol{p_j}$ are attracted towards $\boldsymbol{x}$ depending on their dissimilarity. Thereby, the rank function is calculated as

$$rk(\boldsymbol{x}, \boldsymbol{p_j}, \boldsymbol{\mathcal{P}}) = \sum_{l \in \mathcal{N}} H(d(\boldsymbol{x}, \boldsymbol{p_j}) - d(\boldsymbol{x}, \boldsymbol{p}_l)) , \tag{4}$$

where $H(z)$ is the Heaviside step function, i.e. $H(z) = 1 \iff z > 0$ and 0 else holds. Considering $\boldsymbol{\mathcal{X}}$ as a medium (gas) and λ as a viscosity-parameter with the property $\lambda \xrightarrow{t \to \infty} 0$ supposed to be valid for the training time t, the resulting dynamic resembles Brownian-particles (the prototypes) in a potential. This view is mathematically described by the energy function of NG:

$$E_{NG} = E(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{P}}, \lambda) = \eta(\lambda) \sum_{j \in \mathcal{N}} \int_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} P(\boldsymbol{x}) \cdot h_\lambda(\boldsymbol{x}, \boldsymbol{p}_j, \boldsymbol{\mathcal{P}}) \cdot (\boldsymbol{x} - \boldsymbol{p}_j)^2 d\boldsymbol{x} , \tag{5}$$

where $P(\boldsymbol{\mathcal{X}})$ is the (unknown) data distribution and $\eta(\lambda)$ is a normalization constant depending on λ. It will be omitted in the rest of this paper for simplicity.

2.1 NG for Time-series Prediction

Moody and Darken (1989) as well as *Hecht-Nielsen (1987)* inspired *Martinetz et al. (1993)* to extend the NG to a predictor for time-series data (**NGTSP**). In this view, given $\mathcal{Y} \subseteq \mathbb{R}$, the goal becomes to approximate the input-output mapping $f(\boldsymbol{x}) = y_{\boldsymbol{x}} \in \mathcal{Y}, \forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ by an ensemble $\Pi = \{\pi_1, ..., \pi_k\}$ of local predictors. The resulting model is a hybrid scheme of a pre-initialized NG and the local predictors. In particular, the predictors π_j are only responsible for the data contained in the corresponding Voronoï-cell V_j using

(1). In this way, each neuron is equipped with an adaptive predictor $\pi_j : \mathbb{R}^n \to \mathbb{R}^m$. This strict locality is one of the major differences to RBFN but adopted from *Hecht-Nielsen (1987)*. According to this latter approach, *Martinetz et al. (1993)* defined the local predictors as linear perceptrons

$$\pi_j(\boldsymbol{x}) = \boldsymbol{w}_j^\top(\boldsymbol{x} - \boldsymbol{p}_j) + b_j \,, \tag{6}$$

depending on the vector difference $\boldsymbol{d}_j = \boldsymbol{x} - \boldsymbol{p}_j$ and a weight vector $\boldsymbol{w}_j \in \mathbb{R}^n$ together with a bias $b_j \in \mathbb{R}$ such that $\pi_j : \mathbb{R}^n \to \mathbb{R}\ \forall \boldsymbol{x} \in V_j$ is valid. Thus, the NG dynamic first determines the placement of the prototypes and afterwards the predictors are optimized by using the cost fucntion

$$E_{NGP} = \sum_{j \in \mathcal{N}} \int_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} P(\boldsymbol{x}) \cdot h_{\hat{\lambda}}(\boldsymbol{x}, \boldsymbol{p}_j, \boldsymbol{\mathcal{P}}) \cdot (\pi_j(\boldsymbol{x}) - y_{\boldsymbol{x}})^2 d\boldsymbol{x} \,, \tag{7}$$

which can be considered as the mean squared error of the approximation. Yet, here the schedule of $\hat{\lambda} \xrightarrow{t \to \infty} 0$ does not necessarily need to match the one of λ.

3 Fuzzy-Labeled Neural Gas

Villmann et al. (2006) adapted the NGTSP for soft classification resulting in Fuzzy-Labeled Neural Gas (**FLNG**). In detail this means, that given classes $\mathcal{C} = \{1, ..., L\}$, each data sample $\boldsymbol{x}$ can be assigned to one of the L classes in an possibilistic manner, yielding $\boldsymbol{c}_{\boldsymbol{x}} = (c_1, ..., c_L)$ with $c_j \in [0, 1]$. However, other than in NGTSP, where the local predictors have the range $R(\pi_j) \subseteq \mathbb{R}$, in FLNG $R(\boldsymbol{\pi}_j) \subseteq [0, 1]^L$ is assumed. Thus, following the prediction steps of NGTSP for a stimulus $\boldsymbol{x}$, the response of FLNG is obtained as

$$\boldsymbol{\pi}_s(\boldsymbol{x}) = \boldsymbol{c}_{\boldsymbol{p}_s} \,, \tag{8}$$

with $s = \nu(\boldsymbol{x}, \boldsymbol{\mathcal{P}})$, which results in assigning the possibilistic class vector of the closest prototype $\boldsymbol{p}_s$ to stimulus $\boldsymbol{x}$.

The resulting cost-function was defined by *Villmann et al. (2006)* as

$$C_{FLNG} = \beta \cdot E_{NG} + (1 - \beta) \cdot E_{FL} \,, \tag{9}$$

with $\beta \in [0, 1]$ as a balancing parameter between the energy function E_{NG} of NG and the error caused by the fuzzy-labeling term E_{FL}. This term can be defined either in a discrete manner by

$$E_{FL} = E_{FLD} = \frac{1}{2} \sum_{j \in \mathcal{N}} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} h_\lambda(\boldsymbol{x}, \boldsymbol{p}_j, \boldsymbol{\mathcal{P}}) \cdot (\boldsymbol{\pi}_j(\boldsymbol{x}) - \boldsymbol{c}_{\boldsymbol{x}})^2 \,, \tag{10}$$

with $h_\lambda(\cdot)$ determining the neighbourhood-cooperativesness or as

$$E_{FL} = E_{FLC} = \frac{1}{2} \sum_{j \in \mathcal{N}} \int_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} P(\boldsymbol{x}) \cdot g_\gamma(\boldsymbol{x}, \boldsymbol{p}_j) \cdot (\boldsymbol{\pi}_j(\boldsymbol{x}) - \boldsymbol{c}_{\boldsymbol{x}})^2 d\boldsymbol{x} \,, \tag{11}$$

for the continuous case. In (11) the neighbourhood-cooperativeness of NG is replaced by $g_\gamma(\cdot)$ which is a smooth function depending on the stimuli, as well as on the prototypes. It was considered to be a Gaussian-kernel depending on the dissimilarity of the prototypes and the stimulus, as well as a sigmoidal approximation of the rank-function $rk(\cdot)$ in (4).

In consequence, the prototypes during training receive an update which is dependent on the error caused by the fuzzy classification, which is in addition to the prototype update of the unsupervised NG. Hence, this scheme yields a representation in terms of a supervised scenario controlled by the weighting factor β.

4 Regression Neural Gas

For our contribution we will combine the supervised ideas of FLNG and the approximation approach of NGTSP, while keeping the advantage of parameter reduction and performance maintenance towards RBFN (*Martinetz et al. (1993)*). However, when considered for regression tasks, NGTSP shows several flaws, which result from the predictor definition in (6). These flaws are mostly related to the distance vector $\boldsymbol{d}_j = (\boldsymbol{x} - \boldsymbol{p}_j)$. In particular,

1. for dense clusters in $\boldsymbol{\mathcal{X}}$ it is likely that $\boldsymbol{d}_j \to \boldsymbol{0}$, yielding:

$$\pi_j(\boldsymbol{x}) \overset{\boldsymbol{d}_j \to \boldsymbol{0}}{=} b_j \,, \tag{12}$$

showing that the prediction for stimuli which are close to the prototype, is likely to be estimated as the offset.

2. prototype-based linear regression models defined in a WTA-scheme (1) induce a symmetry, that is assuming $\boldsymbol{x}_1, \boldsymbol{x}_2 \in V_j$ and $\boldsymbol{x}_l = \boldsymbol{p}_j + \alpha_l \cdot \boldsymbol{t}$ for $l \in \{1, 2\}$, i.e. we can draw a line through the prototype $\boldsymbol{p}_j$ connecting $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. However, due to the definition in (6) $\boldsymbol{d}_j$ causes the prediction for such a case to be mainly compensated by the offset b_j. Thus, considering the prediction of $\boldsymbol{x}_l$, we obtain

$$\begin{aligned} \pi_j(\boldsymbol{x}_l) &= \boldsymbol{w}_j^\top (\boldsymbol{x}_l - \boldsymbol{p}_j) + b_j & (13)\\ &= \boldsymbol{w}_j^\top \left[(\boldsymbol{p}_j + \alpha_l \cdot \boldsymbol{t}) - \boldsymbol{p}_j\right] + b_j & (14)\\ &= \alpha_l \cdot \boldsymbol{w}_j^\top \boldsymbol{t} + b_j \,. & (15) \end{aligned}$$

Now, since $\boldsymbol{x}_1, \boldsymbol{x}_2$ are connected by a line going through $\boldsymbol{p}_j$, we can consider the case $-\alpha_1 \approx \alpha_2$, yielding for the respective predictions

$$\pi_j(\boldsymbol{x}_1) = \alpha_1 \cdot \boldsymbol{w}_j^\top \boldsymbol{t} + b_j \,, \tag{16}$$

and for $\boldsymbol{x}_2$

$$\begin{aligned} \pi_j(\boldsymbol{x}_2) &= \alpha_2 \cdot \boldsymbol{w}_j^\top \boldsymbol{t} + b_j & (17)\\ &\approx -\alpha_1 \cdot \boldsymbol{w}_j^\top \boldsymbol{t} + b_j \,, & (18) \end{aligned}$$

showing that such predictions are nearly (depending on $\approx$) symmetric at the point b_j and differ in sign for $b_j = 0$. However, combining 1. and 2. gives that an good estimate for b_j is crucial for the performance of NGTSP, which when considered for regression tasks must not be sufficient, taking into account, that the placement of the prototypes is not influenced by the approximation.

Following the above considerations, we drop the defintion in (6) for a regression model and define instead

$$\tilde{\pi}_j(\boldsymbol{x}) = \boldsymbol{w}_j^\top \tilde{\boldsymbol{x}} + b_j \,, \tag{19}$$

where $\tilde{\boldsymbol{x}}$ can be the stimulus $\boldsymbol{x}$ itself or any transformation $\tilde{\boldsymbol{x}} = T(\boldsymbol{x})$ (e.g. a polynomial transformation) allowing also for non-linear regression architectures and thus for more flexibility.
Note, that in the trivial case $\tilde{\boldsymbol{x}} = \boldsymbol{x}$, the symmetry consideration can still be applied, however expressing $\boldsymbol{x} = \boldsymbol{p}_j + \alpha_{\boldsymbol{x}} \cdot \boldsymbol{t}$, yields

$$\pi_j(\boldsymbol{x}) = \boldsymbol{w}_j^\top \boldsymbol{p}_j + \alpha_{\boldsymbol{x}} \cdot \boldsymbol{w}_j^\top \boldsymbol{t} + b_j \,, \tag{20}$$

where we find that a differing sign is further compensated by $\boldsymbol{p}_j$ itself.
Furthermore, note that $\tilde{\boldsymbol{d}}_j = T(\boldsymbol{d}_j)$ is also possible, however care needs to be taken when chosing $T(\cdot)$, since $\tilde{\boldsymbol{d}}_j \to \boldsymbol{0}$ in 1. might be aggravated.

Moving on with the definition in (19) and combining this with the supervised approach of FLNG, we arrive at the outline of the cost function for a model we call *Regression Neural Gas* (**RegNG**)

$$E_{RegNG} = \beta \cdot E_{NG} + (1-\beta) \cdot E_{Reg} \,, \tag{21}$$

with E_{Reg} as the regression error caused by the current prototype placement, which is defined as

$$E_{Reg} = \sum_{j\in\mathcal{N}} \int_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} P(\boldsymbol{x}) \cdot g_\gamma(\boldsymbol{x}, \boldsymbol{p}_j) \cdot (\tilde{\pi}_j(\boldsymbol{x}) - y_{\boldsymbol{x}})^2 d\boldsymbol{x} \,. \tag{22}$$

Which can be interpreted as an supervised alternative to NGTSP.
Yet another approach is to additonally modify the NG dynamic in terms of the prediction. For this we alternate the rank function (4) of NG to be dependent on the predictors than on the prototypes, which we call *regression-sensitive* ranking rk_{RS}. Considering an arbitrary $\tilde{\pi}_r$ yields

$$rk_{RS}(\boldsymbol{x}, \tilde{\pi}_r, \Pi) = \sum_{j\in\mathcal{N}} H\left(\rho_r(\boldsymbol{x}) - \rho_j(\boldsymbol{x})\right) \,, \tag{23}$$

with

$$\rho_j(\boldsymbol{x}) = (\tilde{\pi}_j(\boldsymbol{x}) - y_{\boldsymbol{x}})^2 \,. \tag{24}$$

Consequently, we replace the function $h_\lambda(\cdot)$ in (3) by

$$h_{RS}(\lambda, \boldsymbol{x}, \tilde{\pi}_r, \Pi) = exp\left(-\frac{rk_{RS}(\boldsymbol{x}, \tilde{\pi}_r, \Pi)}{\lambda}\right) \,. \tag{25}$$

We will refer to this model as *Regression-Sensitive Neural Gas* (**RegSeNG**) describing the cost function

$$E_{RegSeNG} = \beta \cdot E_{RSNG} + (1-\beta) \cdot E_{Reg} \tag{26}$$

where E_{RSNG} describes the NG energy function (5) in which the neighbourhood-function (3) is replaced by (25).

5 Conclusion

In this work we provided new techniques to combine regression related tasks with prototypical frameworks. Such models are often more sparse in terms of number of weights, since the approximation task is subdivided into smaller problems, due to the locality of vector quantization models.

Subsequent work will rely on experiments as well as further extensions like the consideration of relevance learning.

References

Grbovic, M., Vucetic, S.: Regression learning vector quantization. In: 2009 Ninth IEEE International Conference on Data Mining, IEEE (dec 2009), doi:10.1109/icdm.2009.145

Hecht-Nielsen, R.: Counter progagation networks. Applied Optics **26**(23), 4979–4984 (December 1987)

Kaden, M., Schubert, R., Mohannazadeh-Bakhtiari, M., Schwarz, L., Villmann, T.: The LVQ-based counter propagation network an interpretable information bottleneck approach. In: Verleysen, M. (ed.) Proceedings of the 29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2021), Bruges (Belgium), pp. 581–586, i6doc.com, Louvain-La-Neuve, Belgium (2021), doi:10.14428/esann/2021.ES2021-88

Kohonen, T.: Self-organizing formation of topologically correct feature maps. Biol. Cyb. **43**(1), 59–69 (1982)

Kohonen, T.: Self-Organizing Maps. Springer Berlin Heidelberg (1995), doi:10.1007/978-3-642-97610-0

Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In: The Thirty-Second AAAI Conferenceon Artificial Intelligence (AAAI-18), pp. 3530–3537, 432 (2018), doi:10.5555/3504035.3504467, URL https://dl.acm.org/doi/pdf/10.5555/3504035.3504467

Lisboa, P., Saralajew, S., Vellido, A., Fernández-Domenech, R., Villmann, T.: The coming of age of interpretable and explainable machine learning models. Neurocomputing **535**, 25–39 (2023), doi:10.1016/j.neucom.2023.02.040

Martinetz, T., Berkovich, S., Schulten, K.: 'neural-gas' network for vector quantization and its application to time-series prediction. IEEE Transactions on Neural Networks **4**(4), 558–569 (jul 1993), doi:10.1109/72.238311

Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Computation **1**(2), 281–294 (jun 1989), doi:10.1162/neco.1989.1.2.281

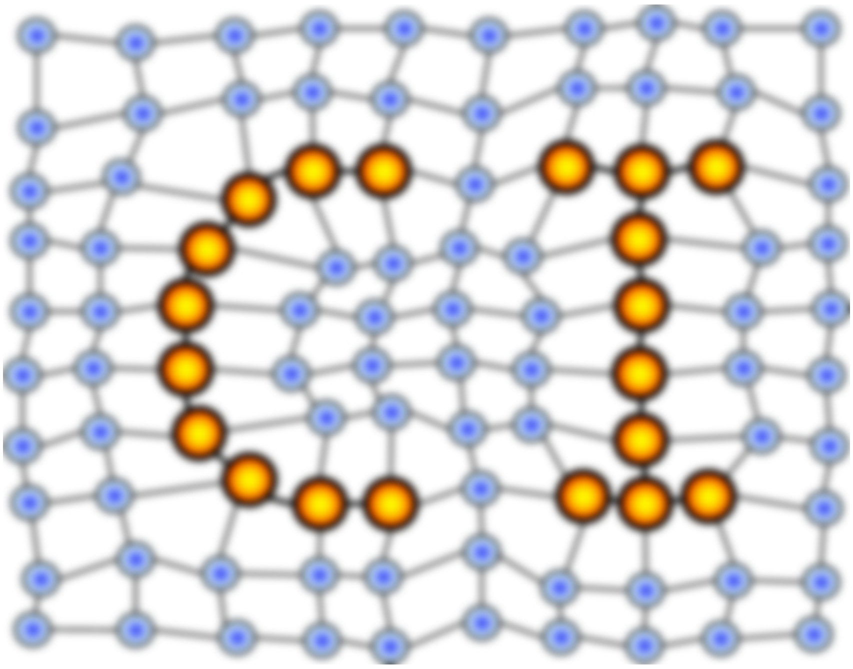
Poggio, T., Girosi, F.: A theory of networks for approximation and learning. Tech. rep., Massachusetts Institute of Technology and Artificial Intellligence Laboratory and Center for Biological Information Processing Whitaker Collage, USA (1989)

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges. Statistics Survey **16**, 1–85 (2022), doi:10.1214/21-SS133

Villmann, T., Hammer, B., Schleif, F., Geweniger, T., Herrmann, W.: Fuzzy classification by fuzzy labeled neural gas. Neural Networks **19**(6-7), 772–779 (jul 2006), doi:10.1016/j.neunet.2006.05.026

Machine Learning Reports

Report 01/2023



Impressum

Machine Learning Reports ISSN: 1865-3960

▽ Publisher/Editors

Prof. Dr. rer. nat. Thomas Villmann
University of Applied Sciences Mittweida
Technikumplatz 17, 09648 Mittweida, Germany
• http://www.mni.hs-mittweida.de/

Dr. rer. nat. Frank-Michael Schleif
University of Bielefeld
Universitätsstrasse 21-23, 33615 Bielefeld, Germany
• http://www.cit-ec.de/tcs/about

▽ Copyright & Licence

▽ Acknowledgments

We would like to thank the reviewers for their time and patience.