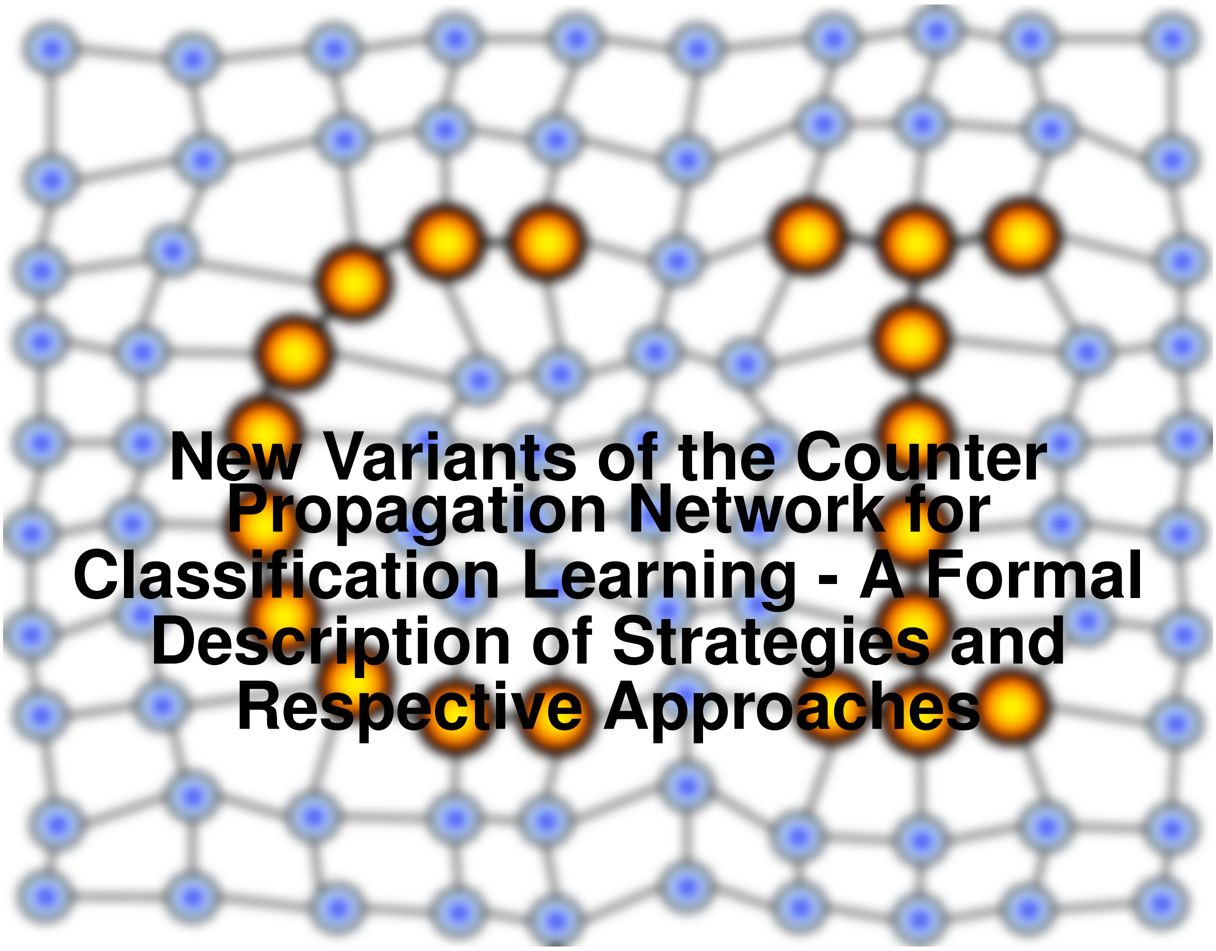
Machine Learning Reports



# New Variants of the Counter Propagation Network for Classification Learning - A Formal Description of Strategies and Respective Approaches

Report 01/2021
Submitted: 25.03.2021
Published: 31.03.2021

T. Villmann [1,*], R. Schubert [1], M. Kaden [1]
(1) Saxon Institute for Computational Intelligence and Machine Learning (SICIM)
Hochschule Mittweida
Mittweida, Germany

**Impressum**

| | |
|---|---|
| Publisher: | University of Applied Sciences Mittweida<br>Technikumplatz 17,<br>09648 Mittweida, Germany |
| Editor: | Prof. Dr. Thomas Villmann<br>Dr. Frank-Michael Schleif |
| Techical-Editor: | Dr. Frank-Michael Schleif |
| Contact: | fschleif@techfak.uni-bielefeld.de |
| URL: | http://techfak.uni-bielefeld.de/$\sim$ fschleif/mlr/mlr.html |
| ISSN: | 1865-3960 |

# New Variants of the Counter Propagation Network fo Classification Learning

## – *A Formal Description of Strategies and Respective Approaches* –

T. Villmann*, R. Schubert and M. Kaden

**University of Applied Sciences Mittweida**

**Saxon Institute for Computational Intelligence and Machine Learning (SICIM)**

**Abstract**

In this paper we present new variants of the counter propagation network introduced already in 1987 by Robert Hecht-Nielsen, which is a network consisting of a vector quantization layer and a subsequent classification layer. In particular, we discuss several vector quantization layers and how to transmit the information to the classification layer. This is discussed in relation to the information-bottleneck-paradigm and regarding perspectives for network training are provided. Thereby, both layers are not longer handled independently during training as in the original approach. More precisely, we explain how the vector quantization layer can be optimized in dependence on the following classification layer. The mathematical formulations of the models are described in detail.

*corresponding author, email: thomas.villmann@hs-mittweida.de

# 1 Introduction and Motivation

The idea of a counter propagation network (CPN), as proposed by R. Hecht-Nielsen [15, 17] in 1987, is to combine a self-organizing map vector quantizer with a perceptron layer for supervised learning. The perceptron layer is also denoted as Grossberg-layer (defined in [13]) in this context and was later also used in ART networks [5]. Recently, the idea of this vector quantization combination with perceptron layers was renewed by C. Rudin for deep multi-layer perceptrons to achieve better robustness [6]. Further, if the vector quantization layer is interpreted as an data compressing tool, we can take a CPN as realization of the information bottleneck method proposed by N. Tishby and N. Zaslavsky in [48, 47]. Furthermore, it can be seen as an approach to the dilemma of representation versus classification as addressed in [34, 36].

The paper first briefly review original CPN. This approach trains the vector quantizer independently from the later Grossberg-layer, which is responsible for the supervised learning task. The main part of the paper deals with several approaches to overcome precisely the information gap. In particular, we investigate how to train the vector quantization layer in dependence of the classification/regression layer. Moreover, we discuss different vector quatizer models instead of the self-organizing map, which is mathematically inconsistent [10, 7]. This includes the neural gas approach from T. Martinetz [31], the Heskes-variant of SOM [19], as well as fuzzy variants of c-means proposed by J. Bezdek [3, 23]. Further, we propose to replace the perceptron layer by variants of learning vector quantizers for classification learning. In fact, we concentrate on classification learning. However, an extension to regression learning is straight-forward.

In this article, we suppose a vector quantizer with reference vectors $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\} \subset \mathbb{R}^n$ for data representation and data $X \subseteq \mathbb{R}^n$. These vectors act as local sensors in the data space to detect signals $\mathbf{x} \in \mathbb{R}^n$ usually by means of a dissimilarity measure $d(\mathbf{x}, \mathbf{w}_k)$. We denote them as *sensoric prototypes* and $W$ is the respective the sensor array (set). Further, we assume data classes $\mathscr{C} = \{1, \ldots, C\}$ and training data labels $c(\mathbf{x}) \in \mathscr{C}$.

# 2 The Original Counter Propagation Network

As already mentioned, the original CPNs consists of two layers. The first one is a self-organizing map (SOM,[26]) layer denoted as in-star layer in this context. The second layer is a perceptron layer called here a Grossberg-outstar-layer [15]. We will denote the first layer as the *vector quantization layer* and the second layer as the *classification layer* to emphasize the more general context later in this paper.

For the SOM we assume that the sensoric prototype set $W$ is related to an external *sensoric grid* $\mathscr{S} \subset \mathbb{R}^p$ according to the feature map model for sensoric data processing introduced by T. Kohonen [24]. More specific, we assume $W$ to be consisting of $K$ prototypes $\mathbf{w}_\mathbf{r} \in W$ where the index $\mathbf{r} \in \mathscr{S}$ refers to a location in the external grid and $k(\mathbf{r}) \in \{1, \ldots, K\}$ returns the respective index in $W$. Usually, the projection dimension $p$ is chosen as $p = 2$ in agreement to cortical areas in human brain [40]. For a given input $\mathbf{x}$ the most appropriate prototype is determined by the winner-takes-all (WTA) rule

$$\mathbf{s}(\mathbf{x}) = \mathrm{argmin}_{\mathbf{r} \in \mathscr{S}} d_E(\mathbf{x}, \mathbf{w}_\mathbf{r}) \tag{1}$$

with $d_E(\mathbf{x}, \mathbf{w}_\mathbf{r})$ being the (squared) Euclidean distance. This WTA-rule is equivalent to the maximum Hebbian excitation principle

$$\mathbf{s}(\mathbf{x}) = \mathrm{argmax}_{\mathbf{r} \in A} \mathbf{w}_\mathbf{r}^T \cdot \mathbf{x}$$

for normalized data and prototypes. We suppose a response vector $\boldsymbol{\xi}(\mathbf{x}, W)$ with $\boldsymbol{\xi}(\mathbf{x}) = (\xi_1, \ldots, \xi_K)^T \in \Xi$ such that the WTA-rule delivers

$$\xi_{k(\mathbf{r})}(\mathbf{x}, W) = \begin{cases} 1 & \mathbf{r} = \mathbf{s}(\mathbf{x}) \\ \\ 0 & \mathbf{r} \neq \mathbf{s}(\mathbf{x}) \end{cases} \tag{2}$$

as stimulus response. We can interpret this mapping $\mathbf{x} \mapsto \boldsymbol{\xi}(\mathbf{x}, W)$ as an information compressing mapping realized by a vector quantizer.

The perceptron layer in CPN consists of a single linear perceptron realized as

$$y(\mathbf{x}) = \boldsymbol{\omega}^T \cdot \boldsymbol{\xi}(\mathbf{x}, W) \tag{3}$$

with an adjustable perceptron weight vector $\boldsymbol{\omega}$. The aim is to adapt this weight vector to predict the class label $c(\mathbf{x})$ as best as possible. In fact, due to the WTA-rule realization (2) the prediction value $y(\mathbf{x})$ is simply obtained according to the weighting $y(\mathbf{x}) = \omega_{k(\mathbf{s}(\mathbf{x}))} \cdot \xi_{k(\mathbf{s}(\mathbf{x}))}(\mathbf{x}, W)$ by means of the respective weight $\omega_{k(\mathbf{s})}$. We can summarize the CPN scheme as

$$X \underset{\text{SOM}}{\longrightarrow} W \underset{\text{crisp}}{\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\longrightarrow}} \Xi \underset{\text{perceptron}}{\overset{y(\mathbf{x})}{\longrightarrow}} \mathscr{C} \tag{4}$$

with the *vector quantization layer (VQ-layer)* $X \underset{\text{SOM}}{\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\longrightarrow}} \Xi$ and the *classification layer (C-layer)* $\Xi \underset{\text{perceptron}}{\overset{y(\mathbf{x})}{\longrightarrow}} \mathscr{C}$.

Training in CPN takes place as in two phases. First, the VQ-layer (SOM) is trained in an unsupervised manner. Second, for the C-layer, the perceptron weight vector $\boldsymbol{\omega}$ is adapted by supervised learning. Thus, the SOM layer yields a grouping of data whereas the perceptron learns to interpret this grouping for classification learning.

Two inprovements are discussed in the community so far:

1. Instead of only one perceptron, class-wise perceptrons are taken [12, p. 186], i.e.

$$y_k(\mathbf{x}) = f\left(\boldsymbol{\omega}_k^T \cdot \boldsymbol{\xi}(\mathbf{x}, W) - \beta_k\right) \tag{5}$$

   with biases $\beta_k$ and an activation function $f$ frequently taken as sigmoid or, currently, promising alternatives like ReLU, swish or others [9, 38, 51].

2. Several authors suggested to relax the WTA-rule taking more than a single winning unit [18, p. 248], [12, p. 189]. For this purpose, $N$ units of the SOM layer surrounding the winning unit $\mathbf{s}$ in $A$ are taken with $\xi_{k(\mathbf{r})}(\mathbf{x}, W) = \frac{1}{N}$ for all these including the winning neuron $\mathbf{s}$.

The CPN approach frequently works very successful although being simple [12, 17, 16, 55, 53, 54, 46, 20, 1]. One can see this also as a historic of incorporation of prototype layers in multi-layer perceptrons as recently discussed in [6]. Yet, the SOM-training takes place independently from the subsequent classification task and, hence, might be suboptimal for the later classification learning. Further, original SOM does not optimize any cost function such that mathematicl guarantees for data grouping behavior are given [10].

In the following we propose new variants and extension of the basic CPN.

# 3 New Variants of the Basic CPN

## 3.1 Modifications for the Vector Quantization Layer and the Response Vector $\boldsymbol{\xi}(\mathbf{x}, W)$

### 3.1.1 Alternatives for the Kohonen-SOM Layer

**3.1.1.1 The Heskes-SOM Layer** The vector quantization layer in original CPN is realized by standard SOM according to T. Kohonen [26]. As already mentioned, the optimization of SOM does not follow a gradient descent scheme of any cost function such that mathematical guarantees cannot be given. Modifying the original winner determination (1) to

$$\mathbf{s}_{\text{Heskes}} = \operatorname{argmin}_{\mathbf{r}} \left( \sum_{\mathbf{r}'} h_\lambda^{\mathscr{S}}(\mathbf{r}', \mathbf{r}) \cdot d(\mathbf{x}, \mathbf{w}_{\mathbf{r}}) \right) \tag{6}$$

yields the Heskes-variant of SOM following a well-defined stochastic gradient descent learning [19], where $\|\mathbf{r}'-\mathbf{r}\|^2_{\mathscr{S}}$ denotes the squared Euclidean distance in the external SOM grid $\mathscr{S}$ and

$$h^{\mathscr{S}}_{\lambda}\left(\mathbf{r}',\mathbf{r}\right)=\exp\left(-\frac{\|\mathbf{r}'-\mathbf{r}\|^2_{\mathscr{S}}}{\lambda}\right) \tag{7}$$

is the SOM *neighborhood function* evaluated for the external sensoric grid $\mathscr{S}$. The prototype update for given input $\mathbf{x}$ is obtained as

$$\Delta\mathbf{w}_{\mathbf{r}}\propto -h^{\mathscr{S}}_{\lambda}\left(\mathbf{s}\left(\mathbf{x}\right),\mathbf{r}\right)\cdot\frac{\partial d\left(\mathbf{x},\mathbf{w}_{\mathbf{r}}\right)}{\partial\mathbf{w}_{\mathbf{r}}} \tag{8}$$

as for the original SOM with $d\left(\mathbf{x},\mathbf{w}_{\mathbf{r}}\right)$ usually being the squared Euclidean distance.

The approach is summarized as

$$X\underset{\text{Heskes-SOM}}{\longrightarrow}W\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\underset{\text{crisp}}{\longrightarrow}}\Xi\overset{y(\mathbf{x})}{\underset{\text{perceptron}}{\longrightarrow}}\mathscr{C} \tag{9}$$

in relation to (4).

**3.1.1.2 The Neural Gas layer** Another alternative to standard SOM is to keep the winner determination but replace the neighborhood function: dropping the external grid $A$ we can define a distance based neighborhood of the prototypes regarding a given input implicitly by the exponential *winning-rank-function*

$$h^{NG}_{\lambda}\left(k,\mathbf{x},W\right)=\exp\left(-\frac{\mathrm{rk}\left(k,\mathbf{x},W\right)}{\lambda}\right) \tag{10}$$

of the prototype $\mathbf{w}_k$. The *rank function* $\mathrm{rk}\left(k,\mathbf{x},W\right)$ is defined in terms of a sum

$$\mathrm{rk}\left(k,\mathbf{x},W\right)=\sum_j H\left(d\left(\mathbf{x},\mathbf{w}_k\right)-d\left(\mathbf{x},\mathbf{w}_j\right)\right) \tag{11}$$

of Haeviside functions

$$H\left(z\right)=\begin{cases}1 & \text{for } z>0\\ \\ 0 & \text{eleswere}\end{cases}$$

such that $\mathrm{rk}\left(s\left(\mathbf{x}\right),\mathbf{x},W\right)=0$ is obtained for the best matching prototype $\mathbf{w}_{s(\mathbf{x})}$. Here, the WTA-rule simply is realized via

$$s\left(\mathbf{x}\right)=\mathrm{argmin}_k d\left(\mathbf{x},\mathbf{w}_k\right) \tag{12}$$

in analogy to the winner determination (1) for SOMs. The prototype dynamic is, similarly as for SOM, obtained as

$$\Delta\mathbf{w}_k\propto -h^{NG}_{\lambda}\left(k,\mathbf{x},W\right)\cdot\frac{\partial d\left(\mathbf{x},\mathbf{w}_k\right)}{\partial\mathbf{w}_k} \tag{13}$$

performing a stochastic gradient descent on the cost function

$$E^{NG}\left(X,W\right)=\frac{1}{C\left(\lambda\right)}\int\sum_k P\left(\mathbf{x}\right)\cdot h^{NG}_{\lambda}\left(k,\mathbf{x},W\right)\cdot d\left(\mathbf{x},\mathbf{w}_k\right)d\mathbf{x} \tag{14}$$

with $C\left(\lambda\right)$ being a normalization constant [31]. In fact, considering the prototypes as gas particles, the dynamic describe the diffusion of the gas according to the data density $P\left(\mathbf{x}\right)$ such that the cost function can be interpreted as the potential function of this gas [31, 30].

The approach can be summarized as

$$X\underset{\text{NG}}{\longrightarrow}W\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\underset{\text{crisp}}{\longrightarrow}}\Xi\overset{y(\mathbf{x})}{\underset{\text{perceptron}}{\longrightarrow}}\mathscr{C} \tag{15}$$

in relation to (4).

#### 3.1.2 Modification of the Response Vector $\boldsymbol{\xi}(\mathbf{x}, W)$

**3.1.2.1 Modifications According to the Vector Quantization Layer** In original CPN, the sensoric response vector $\boldsymbol{\xi}(\mathbf{x}, W)$ contains zeros for all entries except that one for the best matching prototype according to (1) or (12). As already mentioned, this strict rule was suggested to relax that also those entries $\xi_{k(\mathbf{r})}(\mathbf{x}, W)$ are considered to be non-zero, which are neighbored to the winner unit $\mathbf{s}$ in the external grid $A$. This could be taken over to the NG-approach considering the first winning ranks.

A simple generalization of this concept would be to take *gradual responses*

$$\xi_{k(\mathbf{r})}^{\mathscr{S}}(\mathbf{x}, W) = h_{\lambda}^{\mathscr{S}}(\mathbf{s}(\mathbf{x}), \mathbf{r}) \tag{16}$$

for the SOM-layer or

$$\xi_{k}^{G}(\mathbf{x}, W) = h_{\lambda}^{NG}(k, \mathbf{x}, W) \tag{17}$$

in case of a NG-layer. For $\lambda \searrow 0$, the gradual responses (16) and (17) realize a winner-takes-all (WTA) rule, i.e. $\xi_k(\mathbf{x}, W) \neq 0 \Longleftrightarrow k = s$ for NG or $k = k(\mathbf{s}(\mathbf{x}))$ in case of SOM. The derivative of the gradual responses are

$$\begin{aligned}
\frac{\partial \xi_{k(\mathbf{r})}^{\mathscr{S}}(\mathbf{x}, W)}{\partial \mathbf{w}_l} &= \frac{\partial h_{\lambda}^{\mathscr{S}}(\mathbf{s}(\mathbf{x}), \mathbf{r})}{\partial \mathbf{w}_l} \\
&= \frac{\partial}{\partial \mathbf{w}_l} \exp\left(-\frac{(\mathbf{s}(\mathbf{x}) - \mathbf{r})^2}{\lambda}\right) \\
&= -2 \cdot \exp\left(-\frac{(\mathbf{s}(\mathbf{x}) - \mathbf{r})^2}{\lambda}\right) \cdot (\mathbf{s}(\mathbf{x}) - \mathbf{r}) \cdot \frac{\partial \mathbf{s}(\mathbf{x})}{\partial \mathbf{w}_l}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \xi_{k}^{G}(\mathbf{x}, W)}{\partial \mathbf{w}_l} &= \frac{\partial h_{\lambda}^{NG}(k, \mathbf{x}, W)}{\partial \mathbf{w}_l} \\
&= \frac{\partial}{\partial \mathbf{w}_l} \exp\left(-\frac{\mathrm{rk}(k, \mathbf{x}, W)}{\lambda}\right) \\
&= -\exp\left(-\frac{\mathrm{rk}(k, \mathbf{x}, W)}{\lambda}\right) \cdot \frac{\partial \mathrm{rk}(k, \mathbf{x}, W)}{\partial \mathbf{w}_l}
\end{aligned}$$

with remaining derivatives $\frac{\partial \mathbf{s}(\mathbf{x})}{\partial \mathbf{w}_l}$ and $\frac{\partial \mathrm{rk}(k, \mathbf{x}, W)}{\partial \mathbf{w}_l}$, respectively. The first one is not feasible whereas the second one yields

$$\begin{aligned}
\frac{\partial \mathrm{rk}(k, \mathbf{x}, W)}{\partial \mathbf{w}_l} &= \frac{\partial \sum_j H(d(\mathbf{x}, \mathbf{w}_k) - d(\mathbf{x}, \mathbf{w}_j))}{\partial \mathbf{w}_l} \\
&= \sum_j \frac{\partial H(d(\mathbf{x}, \mathbf{w}_k) - d(\mathbf{x}, \mathbf{w}_j))}{\partial \mathbf{w}_k} \\
&= \sum_j \frac{\partial H(d(\mathbf{x}, \mathbf{w}_k) - d(\mathbf{x}, \mathbf{w}_j))}{d(\mathbf{x}, \mathbf{w}_k)} \cdot \frac{\partial d(\mathbf{x}, \mathbf{w}_k)}{\partial \mathbf{w}_l} \\
&= \delta_{k,l} \cdot \frac{\partial d(\mathbf{x}, \mathbf{w}_k)}{\partial \mathbf{w}_k} \cdot \sum_j \frac{\partial H(d(\mathbf{x}, \mathbf{w}_k) - d(\mathbf{x}, \mathbf{w}_j))}{d(\mathbf{x}, \mathbf{w}_k)} \\
&= \delta_{k,l} \cdot \frac{\partial d(\mathbf{x}, \mathbf{w}_k)}{\partial \mathbf{w}_k} \cdot \sum_j \delta_{\mathrm{Dirac}}(d(\mathbf{x}, \mathbf{w}_k) - d(\mathbf{x}, \mathbf{w}_j))
\end{aligned} \tag{18}$$

where $\delta_{k,l}$ denotes the Kronecker symbol, i.e. $\delta_{k,l} = 1 \Longleftrightarrow k = l$ and it is zero else.

However, these gradual responses ignore the the distance values. Therefore, we suggest to consider the *local responses*

$$\xi_{k(\mathbf{r})}^{SOM}(\mathbf{x},W)=h_{\lambda}^{\mathscr{S}}(\mathbf{s}(\mathbf{x}),\mathbf{r})\cdot d(\mathbf{x},\mathbf{w_r}) \tag{19}$$

and

$$\xi_{k}^{NG}(\mathbf{x},W)=h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot d(\mathbf{x},\mathbf{w}_k) \tag{20}$$

for SOM- and NG-layer, respectively. These local responses reflect both the winning rank as well as the dissimilarity. This is consistent to the winner determination (6) of the Heskes-SOM.

Moreover, the expected NG-response is

$$\left\langle\xi_{k}^{NG}(\mathbf{x},W)\right\rangle_{\mathbf{x}}=\int P(\mathbf{x})\cdot h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot d(\mathbf{x},\mathbf{w}_k)\,d\mathbf{x} \tag{21}$$

such that the sum $\sum_k\langle\xi_k(\mathbf{x},W)\rangle_{\mathbf{x}}$ is equivalent to the energy function (14) of the neural gas vector quantizer [31], if we swap integration and summation.

Yet, for the (non-averaged) gradient we have

$$\begin{aligned}
\frac{\partial\xi_{k}^{NG}(\mathbf{x},W)}{\partial\mathbf{w}_l} &= \frac{\partial}{\partial\mathbf{w}_l}\left(h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot d(\mathbf{x},\mathbf{w}_k)\right)\\
&= \frac{\partial}{\partial\mathbf{w}_l}\left(h_{\lambda}^{NG}(k,\mathbf{x},W)\right)\cdot d(\mathbf{x},\mathbf{w}_k)+\delta_{k,l}\cdot h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\\
&\overset{(10)}{=} -h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot\frac{1}{\lambda}\cdot\frac{\partial\mathrm{rk}(k,\mathbf{x},W)}{\partial\mathbf{w}_l}\cdot d(\mathbf{x},\mathbf{w}_k)+\delta_{k,l}\cdot h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\\
&= -h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\lambda}\cdot\frac{\partial\mathrm{rk}(k,\mathbf{x},W)}{\partial\mathbf{w}_l}+\delta_{k,l}\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\right)\\
&\overset{(18)}{=} -h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\lambda}\cdot\delta_{k,l}\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\cdot\sum_j\delta_{\mathrm{Dirac}}\left(d(\mathbf{x},\mathbf{w}_k)-d(\mathbf{x},\mathbf{w}_j)\right)+\delta_{k,l}\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\right) && (22)\\
&= -h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot\delta_{k,l}\cdot\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\lambda}\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\cdot\sum_j\delta_{\mathrm{Dirac}}\left(d(\mathbf{x},\mathbf{w}_k)-d(\mathbf{x},\mathbf{w}_j)\right)+\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\right) && (23)\\
& && (24)\\
&\overset{\text{in prob.}}{=} -h_{\lambda}^{NG}(k,\mathbf{x},W)\cdot\delta_{k,l}\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k}\cdot\frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial\mathbf{w}_k} && (25)
\end{aligned}$$

supposing a data density $P(\mathbf{x})$ with $\forall\mathbf{x}:P(\mathbf{x})<\infty$ to be valid.

The approach is summarized as

$$X\underset{\text{NG/SOM}}{\longrightarrow}W\underset{\text{NG/SOM-like}}{\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\longrightarrow}}\Xi\underset{\text{perceptron}}{\overset{y(\mathbf{x})}{\longrightarrow}}\mathscr{C} \tag{26}$$

in relation to (4).

#### 3.1.2.2 Modifications According to a Fuzzy Vector Quantization Interpretation

Another choice for gradual sensoric responses for a given prototype set $W$ is to evaluate fuzzy assignments of the data. According to [2, 3], the fuzzy assignment

$$u_k(\mathbf{x})=\frac{1}{\sum_{j=1}^{K}\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)^{\frac{2}{m-1}}} \tag{27}$$

gives the probability that data point $\mathbf{x}$ is assigned to prototype $\mathbf{w}_k$ taken as a local cluster center according to the fuzzy-c-means approach (FCM,[22]). The parameter $m>1$ is the fuzzyfier usually chosen as $m=2$. Explicitly note that $\sum_{j=1}^{K} u_j(\mathbf{x}) = 1$ is valid for the fuzzy assignments. Yet, the fuzzy assignments do not reflect all aspects of cluster assignments, in particular, if the distance $d(\mathbf{x}, \mathbf{w}_k)$ is large. Therefore, the typicality assignments

$$t_k(\mathbf{x}) = \frac{1}{1 + \left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\gamma_k}\right)^{\frac{1}{m-1}}} \tag{28}$$

were proposed [27] with the normalization

$$\gamma_k = \gamma \cdot \frac{E_{\mathbf{x}}[u_k(\mathbf{x}) \cdot d(\mathbf{x}, \mathbf{w}_k)]}{E_{\mathbf{x}}[u_k(\mathbf{x})]}$$

and the usual choice $\gamma = 1$ [28, 35]. Here, $E_{\mathbf{x}}[\cdot]$ denotes the expectation operator with respect to $\mathbf{x}$.

Both quantities, the fuzzy and the typicality assignments, can be combined by a convex sum

$$\xi_k^F(\mathbf{x}, W) = \alpha \cdot u_k(\mathbf{x}) + (1 - \alpha) \cdot t_k(\mathbf{x}) \tag{29}$$

with $\alpha \in (0, 1)$ to obtain fuzzy-based sensoric responses.

The gradient $\frac{\partial u_k(\mathbf{x})}{\partial \mathbf{w}_l}$ of the fuzzy assignments is calculated as

$$\begin{aligned}
\frac{\partial u_k(\mathbf{x})}{\partial \mathbf{w}_l} &= \frac{-1}{\left(\sum_{j=1}^{K}\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)^{\frac{2}{m-1}}\right)^2} \cdot \frac{\partial \sum_{j=1}^{K}\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)^{\frac{2}{m-1}}}{\partial \mathbf{w}_l} \\
&= \frac{-1}{\left(\sum_{j=1}^{K}\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)^{\frac{2}{m-1}}\right)^2} \cdot \sum_{j=1}^{K} \frac{\partial \left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)^{\frac{2}{m-1}}}{\partial \mathbf{w}_l} \\
&= \frac{-1}{\left(\sum_{j=1}^{K}\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)^{\frac{2}{m-1}}\right)^2} \cdot \sum_{j=1}^{K} \left(\frac{2}{m-1} \cdot \left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)^{\frac{1-m}{m-1}} \cdot \frac{\partial}{\partial \mathbf{w}_l}\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right)\right)
\end{aligned}$$

with

$$\frac{\partial}{\partial \mathbf{w}_l}\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{d(\mathbf{x},\mathbf{w}_j)}\right) = \begin{cases} \frac{1}{d(\mathbf{x},\mathbf{w}_j)} \cdot \frac{\partial d(\mathbf{x},\mathbf{w}_l)}{\partial \mathbf{w}_l} & l = k \neq j \\ \frac{-d(\mathbf{x},\mathbf{w}_k)}{(d(\mathbf{x},\mathbf{w}_l))^2} \cdot \frac{\partial d(\mathbf{x},\mathbf{w}_l)}{\partial \mathbf{w}_l} & l = j \neq k \\ 0 & else \end{cases}$$

whereas

$$\begin{aligned}
\frac{\partial t_k(\mathbf{x})}{\partial \mathbf{w}_l} &= \frac{-1}{\left(1 + \left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\gamma_k}\right)^{\frac{1}{m-1}}\right)^2} \cdot \frac{\partial}{\partial \mathbf{w}_l}\left(\left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\gamma_k}\right)^{\frac{1}{m-1}}\right) \\
&= \frac{-1}{\left(1 + \left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\gamma_k}\right)^{\frac{1}{m-1}}\right)^2} \cdot \frac{1}{m-1} \cdot \left(\frac{d(\mathbf{x},\mathbf{w}_k)}{\gamma_k}\right)^{\frac{2-m}{m-1}} \cdot \frac{1}{\gamma_k} \cdot \frac{\partial d(\mathbf{x},\mathbf{w}_k)}{\partial \mathbf{w}_l}
\end{aligned}$$

is obtained for the typicalities. Thus, the derivative

$$\frac{\partial \xi_k^F(\mathbf{x}, W)}{\partial \mathbf{w}_l} = \alpha \cdot \frac{\partial u_k(\mathbf{x})}{\partial \mathbf{w}_l} + (1 - \alpha) \cdot \frac{\partial t_k(\mathbf{x})}{\partial \mathbf{w}_l} \tag{30}$$

describes gradient of this fuzzy response. Yet, other fuzzy approaches like the application of general t-norms could also be of interest [11].

The approach is summarized as

$$X \underset{\text{FCM}}{\longrightarrow} W \underset{\text{fuzzy}}{\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\longrightarrow}} \Xi \underset{\text{perceptron}}{\overset{y(\mathbf{x})}{\longrightarrow}} \mathscr{C} \tag{31}$$

in relation to (4).

## 3.2 Modifications of the Classification Layer

As mentioned in the beginning, in original CPN, the classification layer consists of one or several perceptrons taking the sensoric reponse vector $\boldsymbol{\xi}(\mathbf{x}, W)$ as input. It is optimized after vector quantization training and does not fine-tune the vector quantization layer. Hence, no backward information is considered. In the following we propose several modifications of that scheme. To keep in mind that the sensoric response vector $\boldsymbol{\xi}(\mathbf{x}, W)$ depends on the

### 3.2.1 Multilayer Perceptron Layer

An obvious way to generalize the CPN is to replace the perceptron(s) in the classification layer by a (deep) multi-layer perceptron architecture with cross-entropy loss as cost function. This would realize the idea to incorporate vector quantization layers into deep networks as suggested in [6]. Considering the response determination as a mapping $\mathbf{x} \mapsto \boldsymbol{\xi}(\mathbf{x}, W)$ realized by a vector quantizer (SOM/NG/FCM), one can think to fine-tune the response mapping $\boldsymbol{\xi}(\mathbf{x}, W)$ by means of the gradients $\frac{\partial \boldsymbol{\xi}(\mathbf{x},W)}{\partial \mathbf{w}_l}$ for stochastic gradient learning with respect to the cross entropy. However, because of the deep architecture, the problem of vanishing gradients becomes apparent for this approach and has to be tackled carefully. Further, the interpretability of the vector quantization layer is destroyed by the subsequent deep network, which counter-acts to one of the central paradigms of vector quantization models.

The approach can be summarized as

$$X \underset{\text{NG/FCM}}{\longrightarrow} W \underset{\text{NG-like/fuzzy}}{\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\longrightarrow}} \Xi \underset{\text{deep network}}{\overset{y(\mathbf{x})}{\longrightarrow}} \mathscr{C} \tag{32}$$

in relation to (4).

### 3.2.2 LVQ Layers

**3.2.2.1 Non-probabilistic LVQ classifier** We suggest to replace the perceptron layer of CPN by a generalized learning vector quantization classifier (GLVQ, [44]) as cost function based variant of the heuristic learning vector quantizer (LVQ) introduced by T. Kohonen [25]. LVQ-models are interpretable prototype-based classifiers relying on an attraction-repulsing scheme for the prototypes [52]. Originally, LVQ was established to approximate Bayesian learning in supervised vector quantization learning for classification [26]. GLVQ is known to be interpretable [4, 50, 21], robust [42, ?] and implicitly optimizes the hypothesis margin during classification learning [8, 43].

Taking the sensoric responses $\boldsymbol{\xi}$ as input, the GLVQ assumes prototypes $\boldsymbol{\omega}_j \in \mathbb{R}^{\mathbb{K}}$ with class labels $c(\boldsymbol{\omega}_j)$. We denote the set $\mathcal{W} = \{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_M\}$ as GLVQ-prototypes to distinguish them from the sensoric prototype set $W$. The cost function optimized by GLVQ-training in our setting is

$$E_{GLVQ}(\mathcal{W}) = \sum_{\mathbf{x}} f(\mu(\boldsymbol{\xi}(\mathbf{x}, W)))$$

where $f$ is a differentiable sigmoid function with range $[0, 1]$ and and the classifier function $\mathcal{W}$-dependent

$$\mu(\boldsymbol{\xi}(\mathbf{x}, W), \mathcal{W}) = \frac{\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^+) - \delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^-)}{\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^+) + \delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^-)} \tag{33}$$

with the dissimilarity measure

$$\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}) = (\boldsymbol{\Omega}(\boldsymbol{\xi}(\mathbf{x}, W) - \boldsymbol{\omega}))^2 \tag{34}$$

known from the matrix GLVQ (GMLVQ) [45] anf relevance GLVQ (GRLVQ) [14]. Thereby, $\boldsymbol{\omega}^+$ is the best matching prototype $\boldsymbol{\omega}_j$ according to the WTA-rule (1) known from SOM with the constraint of class label agreement $c(\boldsymbol{\omega}_j) = c(\boldsymbol{\xi}(\mathbf{x}, W))$. Analogously, $\boldsymbol{\omega}^-$ is the best matching prototype among all prototypes responsible for other classes than $c(\boldsymbol{\xi}(\mathbf{x}, W))$. Thus, the classifier function $\mu(\boldsymbol{\xi}(\mathbf{x}, W), \mathcal{W}) \in [-1, 1]$ becomes negative for correct classification. After training the network response is the label $c(\boldsymbol{\omega}_s)$ of the overall best matching prototype $\boldsymbol{\omega}_s$ according to the WTA-rule (1).

Stochastic gradient descent learning on the cost function $E_{GLVQ}(\mathcal{W})$ takes place as prototype updates according to the derivative of the local errors

$$E_{GLVQ}(\mathbf{x}, \mathcal{W}, W) = f(\mu(\boldsymbol{\xi}(\mathbf{x}, W), \mathcal{W}))$$

i.e.

$$\Delta\boldsymbol{\omega}^{\pm} \propto -\frac{\partial E_{GLVQ}(\mathbf{x}, \mathcal{W}, W)}{\partial\boldsymbol{\omega}^{\pm}}$$

has to be considered. In the following, we omit for $\boldsymbol{\xi}(\mathbf{x}, W)$ the dependencies on $\mathbf{x}$ and $W$ for simplicity if it is not necessary to refer explicitly to the dependencies. Thus, the gradient formally read as

$$\frac{\partial E_{GLVQ}(\mathbf{x}, \mathcal{W}, W)}{\partial\boldsymbol{\omega}^{\pm}} = \frac{\partial f(\mu)}{\partial\mu(\boldsymbol{\xi})} \cdot \frac{\partial\mu(\boldsymbol{\xi})}{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{\pm})} \cdot \frac{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{\mp})}{\partial\boldsymbol{\omega}^{\pm}} \tag{35}$$

with

$$\frac{\partial\mu(\boldsymbol{\xi})}{\partial\boldsymbol{\xi}} = \frac{\partial\mu(\boldsymbol{\xi})}{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{+})} \cdot \frac{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{+})}{\partial\boldsymbol{\xi}} + \frac{\partial\mu(\boldsymbol{\xi})}{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{-})} \cdot \frac{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{-})}{\partial\boldsymbol{\xi}} \tag{36}$$

for $\boldsymbol{\omega}^+, \boldsymbol{\omega}^-$. Further, we get

$$\frac{\partial\mu(\boldsymbol{\xi})}{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{\pm})} = \frac{\mp 2 \cdot \delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{\mp})}{(\delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{+}) + \delta(\boldsymbol{\xi}, \boldsymbol{\omega}^{-}))^2}$$

as derivatives for $\mu(\boldsymbol{\xi})$ depending on $\boldsymbol{\omega}^+$ and $\boldsymbol{\omega}^-$. The gradients

$$\nabla_{\boldsymbol{\xi}}\delta(\boldsymbol{\xi}, \boldsymbol{\omega}) = \frac{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega})}{\partial\boldsymbol{\xi}} = 2 \cdot \boldsymbol{\Omega}^T\boldsymbol{\Omega}(\boldsymbol{\xi} - \boldsymbol{\omega})$$

and

$$\nabla_{\boldsymbol{\omega}}\delta(\boldsymbol{\xi}, \boldsymbol{\omega}) = \frac{\partial\delta(\boldsymbol{\xi}, \boldsymbol{\omega})}{\partial\boldsymbol{\omega}} = -2 \cdot \boldsymbol{\Omega}^T\boldsymbol{\Omega}(\boldsymbol{\xi} - \boldsymbol{\omega})$$

reflect the contribution of the dissimilarity measure $\delta(\boldsymbol{\xi}, \boldsymbol{\omega})$.

Because the sensoric inputs $\boldsymbol{\xi}(\mathbf{x}, W)$ depend on the sensoric prototypes $\mathbf{w}_k$, we can optimize the classification performance of the GLVQ model also with respect to these quantities. Thus we have to consider the derivative

$$\begin{aligned}\frac{\partial\mu(\boldsymbol{\xi}(\mathbf{x}, W))}{\partial\mathbf{w}_k} &= \frac{\partial\mu(\boldsymbol{\xi}(\mathbf{x}, W))}{\partial\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^{+})} \cdot \frac{\partial\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^{+})}{\partial\boldsymbol{\xi}(\mathbf{x}, W)} \cdot \frac{\partial\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^{+})}{\partial\mathbf{w}_k} + \\ &\quad + \frac{\partial\mu(\boldsymbol{\xi})}{\partial\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^{-})} \cdot \frac{\partial\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^{-})}{\partial\boldsymbol{\xi}} \cdot \frac{\partial\delta(\boldsymbol{\xi}(\mathbf{x}, W), \boldsymbol{\omega}^{-})}{\partial\mathbf{w}_k}\end{aligned} \tag{37}$$

using (36). For the NG-like responses we calculate

$$\begin{aligned}
\frac{\partial\delta\left(\boldsymbol{\xi}^{NG}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)}{\partial\mathbf{w}_k} &= \left(\nabla_{\boldsymbol{\xi}}\delta\left(\mathbf{x},\boldsymbol{\omega}\right)\right)^T\cdot\frac{\partial\boldsymbol{\xi}^{NG}}{\partial\mathbf{w}_k} \\
&= \sum_{l=1}^{K}\left(\nabla_{\boldsymbol{\xi}}\delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)\right)_l\cdot\frac{\partial\xi_l^{NG}\left(\mathbf{x},W\right)}{\partial\mathbf{w}_k} \\
&\overset{(22)}{=} -\sum_{l=1}^{K}\left(\nabla_{\boldsymbol{\xi}}\delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)\right)_l\cdot\exp\left(-\frac{\mathrm{rk}\left(k,\mathbf{x},W\right)}{\lambda}\right) \\
&\quad\cdot\delta_{k,l}\cdot\left(\frac{d\left(\mathbf{x},\mathbf{w}_k\right)}{\lambda}\cdot\frac{\partial d\left(\mathbf{x},\mathbf{w}_k\right)}{\partial\mathbf{w}_k}\cdot\sum_j\delta_{\mathrm{Dirac}}\left(d\left(\mathbf{x},\mathbf{w}_k\right)-d\left(\mathbf{x},\mathbf{w}_j\right)\right)+\frac{\partial d\left(\mathbf{x},\mathbf{w}_k\right)}{\partial\mathbf{w}_k}\right) && (38) \\
& && (39) \\
&= -\left(\nabla_{\boldsymbol{\xi}}\delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)\right)_k\cdot\exp\left(-\frac{\mathrm{rk}\left(k,\mathbf{x},W\right)}{\lambda}\right) \\
& && (40) \\
&\quad\cdot\left(\frac{d\left(\mathbf{x},\mathbf{w}_k\right)}{\lambda}\cdot\frac{\partial d\left(\mathbf{x},\mathbf{w}_k\right)}{\partial\mathbf{w}_k}\cdot\sum_j\delta_{\mathrm{Dirac}}\left(d\left(\mathbf{x},\mathbf{w}_k\right)-d\left(\mathbf{x},\mathbf{w}_j\right)\right)+\frac{\partial d\left(\mathbf{x},\mathbf{w}_k\right)}{\partial\mathbf{w}_k}\right) && (41) \\
&\overset{\text{in prob. (25)}}{=} -\left(\nabla_{\boldsymbol{\xi}}\delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)\right)_k\cdot\exp\left(-\frac{\mathrm{rk}\left(k,\mathbf{x},W\right)}{\lambda}\right)\cdot\frac{\partial d\left(\mathbf{x},\mathbf{w}_k\right)}{\partial\mathbf{w}_k} && (42)
\end{aligned}$$

to be used in (37).

For the fuzzy responses $\xi_k^F\left(\mathbf{x},W\right)$ from (29) we similarly get (30)

$$\begin{aligned}
\frac{\partial\delta\left(\boldsymbol{\xi}^{F}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)}{\partial\mathbf{w}_k} &= \left(\nabla_{\boldsymbol{\xi}}\delta\left(\mathbf{x},\boldsymbol{\omega}\right)\right)^T\cdot\frac{\partial\boldsymbol{\xi}^{F}}{\partial\mathbf{w}_k} \\
&= \sum_{l=1}^{K}\left(\nabla_{\boldsymbol{\xi}}\delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)\right)_l\cdot\frac{\partial\xi_l^{F}\left(\mathbf{x},W\right)}{\partial\mathbf{w}_k} \\
&\overset{(30)}{=} \sum_{l=1}^{K}\left(\nabla_{\boldsymbol{\xi}}\delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}\right)\right)_l\cdot\left(\alpha\cdot\frac{\partial u_l\left(\mathbf{x}\right)}{\partial\mathbf{w}_k}+\left(1-\alpha\right)\cdot\frac{\partial t_l\left(\mathbf{x}\right)}{\partial\mathbf{w}_k}\right) && (43)
\end{aligned}$$

This GMLVQ-approach (and its variants) can be summarized as

$$X \underset{\text{NG/FCM}}{\leftrightarrows} W \underset{\text{NG-like/fuzzy}}{\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\leftrightarrows}} \Xi \underset{\text{GMLVQ}}{\overset{c(\boldsymbol{\omega}_s)}{\longrightarrow}} \mathscr{C} \tag{44}$$

in relation to (4).

**3.2.2.2 Probabilistic LVQ classifier** Probabilistic LVQ (PLVQ,[49]) uses information theoretic concepts to estimate the model probabilities $p_{\mathcal{W}}\left(c|\boldsymbol{\xi}\right)$. Let $\mathbf{p}\left(\boldsymbol{\xi}\right)=\left(p_1\left(\boldsymbol{\xi}\right),\ldots,p_C\left(\boldsymbol{\xi}\right)\right)$ be the class probability vector for sample $\boldsymbol{\xi}$ and

$$\mathbf{p}_{\mathcal{W}}\left(\boldsymbol{\xi}\right)=\left(p_{\mathcal{W}}\left(1|\boldsymbol{\xi}\right),\ldots,p_{\mathcal{W}}\left(C|\boldsymbol{\xi}\right)\right) \tag{45}$$

the respective *predicted class probability vector* provided by the probabilistic classifier model depending on PLVQ-prototype set $\mathcal{W}=\left\{\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_M\right\}$ as before. The target labels for trainng data are denoted by $\mathbf{t}\left(\boldsymbol{\xi}\right)\in\left[0,1\right]^C$ describing a probabilistic class assignment according to $t_c\left(\boldsymbol{\xi}\right)\in\left[0,1\right]$, i.e. $\sum_c t_c\left(\boldsymbol{\xi}\right)=1$. For a possibilistic assignment, the latter constraint has to be dropped.

The mutual information between $\mathbf{t}(\boldsymbol{\xi})$ and $\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi})$ has to be maximized, and, hence, the corresponding KLD $D_{KL}(\mathbf{p}(\boldsymbol{\xi})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}))$ has to be minimized for the probabilistic setting, which is equivalent to maximize the cross-entropy

$$Cr(\mathbf{t}(\boldsymbol{\xi})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi})) = \sum_c t_c(\boldsymbol{\xi})\cdot\log\left(p_{\mathcal{W}}(c|\boldsymbol{\xi})\right) \tag{46}$$

as shown in [37, p. 221ff]. For PLVQ, the cross-entropy $Cr(\mathbf{t}(\boldsymbol{\xi})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}))$ plays the role of a local cost such that

$$L_{PLVQ}(X,\mathcal{W}) = -\sum_k Cr(\mathbf{t}(\boldsymbol{\xi}_k)\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}_k)) \tag{47}$$

has to be minimzed [49]. Alternatively, the loss

$$L^{\alpha}_{PLVQ}(X,\mathcal{W}) = \sum_k D_{\alpha}(\mathbf{t}(\boldsymbol{\xi}_k)\,||p_{\mathcal{W}}(\boldsymbol{\xi}_k)) \tag{48}$$

based on the Rényi divergence [39]

$$D_{\alpha}((\boldsymbol{\xi}_k)\,||p_{\mathcal{W}}(\boldsymbol{\xi}_k)) = \frac{1}{1-\alpha}\log\left(\sum_c (t_c(\boldsymbol{\xi}_k))^{\alpha}\cdot(p_{\mathcal{W}}(c|\boldsymbol{\xi}_k))^{1-\alpha}\right) \tag{49}$$

could be taken as cost function. For the cross-entropy $Cr(\mathbf{t}(\boldsymbol{\xi})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}))$, the conditional class probability $p_{\mathcal{W}}(c|\boldsymbol{\xi}_k)$ depends on $p(\boldsymbol{\xi}|\mathbf{w}_j)$ via the class prediction probabilities

$$\begin{aligned}
p_{\mathcal{W}}(c|\boldsymbol{\xi}) &= \frac{P_{\mathcal{W}}(\boldsymbol{\xi},c)}{P_{\mathcal{W}}(\boldsymbol{\xi})}\\
&= \frac{\sum_{j:c(\boldsymbol{\omega}_j)=c} p(\boldsymbol{\xi}|\boldsymbol{\omega}_j)\cdot p(\boldsymbol{\omega}_j)}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)}\\
&= \sum_{j:c(\boldsymbol{\omega}_j)=c} S_{\mathcal{W}}(j,\boldsymbol{\xi})
\end{aligned} \tag{50}$$

with

$$S_{\mathcal{W}}(j,\boldsymbol{\xi}) = \frac{p(\boldsymbol{\xi}|\boldsymbol{\omega}_j)\cdot p(\boldsymbol{\omega}_j)}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)} \tag{51}$$

as so-called local quantities. For the possibilistic setting we refer to [32].

The gradient of the cross-entropy $Cr(\mathbf{t}(\mathbf{x})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}))$ from (46) with respect to the PLVQ-prototypes $\boldsymbol{\omega}_l$ reads as

$$\begin{aligned}
\frac{\partial Cr(\mathbf{t}(\boldsymbol{\xi})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}))}{\partial\boldsymbol{\omega}_l} &= \frac{\partial}{\partial\boldsymbol{\omega}_l}\left(\sum_c t_c(\boldsymbol{\xi})\cdot\log\left(p_{\mathcal{W}}(c|\boldsymbol{\xi})\right)\right)\\
&= \sum_c \frac{t_c(\boldsymbol{\xi})}{p_{\mathcal{W}}(c|\boldsymbol{\xi})}\cdot\frac{\partial p_{\mathcal{W}}(c|\boldsymbol{\xi})}{\partial\boldsymbol{\omega}_l}\\
&\overset{(50)}{=} \sum_c \frac{t_c(\boldsymbol{\xi})}{p_{\mathcal{W}}(c|\boldsymbol{\xi})}\cdot\frac{\partial}{\partial\boldsymbol{\omega}_l}\left(\sum_{j:c(\mathbf{w}_j)=c} S_{\mathcal{W}}(j,\boldsymbol{\xi})\right)\\
&= \sum_c \frac{t_c(\boldsymbol{\xi})}{p_{\mathcal{W}}(c|\boldsymbol{\xi})}\cdot\left(\sum_{j:c(\mathbf{w}_j)=c}\frac{\partial S_{\mathcal{W}}(j,\boldsymbol{\xi})}{\partial\boldsymbol{\omega}_l}\right)
\end{aligned}$$

with

$$
\begin{aligned}
\frac{\partial S_{\mathcal{W}}(j,\boldsymbol{\xi})}{\partial \boldsymbol{\omega}_l} &= \frac{\partial}{\partial \boldsymbol{\omega}_l}\left(\frac{p(\boldsymbol{\xi}|\boldsymbol{\omega}_j)\cdot p(\boldsymbol{\omega}_j)}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)}\right) \\
&= \frac{\delta_{kl}}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)} - p(\boldsymbol{\xi}|\boldsymbol{\omega}_j)\, p(\boldsymbol{\omega}_j)\left(\frac{\frac{\partial p(\boldsymbol{\xi}|\boldsymbol{\omega}_l)\cdot p(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}_l}}{\left(\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)\right)^2}\right) \\
&= \frac{\delta_{kl}}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)} - S_{\mathcal{W}}(j,\boldsymbol{\xi})\cdot\left(\frac{\frac{\partial p(\boldsymbol{\xi}|\boldsymbol{\omega}_l) p(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}_l}}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)}\right) \\
&= \frac{\delta_{kl} - S_{\mathcal{W}}(j,\boldsymbol{\xi})\cdot \frac{\partial p(\boldsymbol{\xi}|\boldsymbol{\omega}_l) p(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}_l}}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)} \\
&= \frac{\delta_{kl} - S_{\mathcal{W}}(j,\boldsymbol{\xi})\cdot \left(p(\boldsymbol{\omega}_l)\cdot\frac{\partial p(\boldsymbol{\xi}|\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}_l} + p(\boldsymbol{\xi}|\boldsymbol{\omega}_l)\cdot\frac{\partial p(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}_l}\right)}{\sum_{k=1}^{N} p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)\cdot p(\boldsymbol{\omega}_k)}
\end{aligned}
$$

according to (51).

Now, we assume an arbitrary non-negative dissimilarity measure $\delta(\boldsymbol{\xi},\boldsymbol{\omega})$ according to [33] such that $\delta : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathscr{D} \subseteq \mathbb{R}_+$, where $\mathscr{D}$ is the *data dissimilarity space*. Further, let the conditional probability $p(\boldsymbol{\xi}|\boldsymbol{\omega})$ be only depending on the measure $\delta(\boldsymbol{\xi},\boldsymbol{\omega})$, i.e.

$$
p(\boldsymbol{\xi}|\boldsymbol{\omega}_j) = \pi_{\mathscr{D}}(\delta(\boldsymbol{\xi},\boldsymbol{\omega}_j))\ . \tag{52}
$$

is an one-dimensional differentiable density function representing $P_{\mathscr{D}}$. We denote $P_{\mathscr{D}}(\delta(\boldsymbol{\xi}_i,\boldsymbol{\omega}_k))$ as a *dissimilarity density model*. Then we get

$$
\frac{\partial p(\boldsymbol{\xi}|\boldsymbol{\omega}_k)}{\partial \boldsymbol{\omega}_l} = \frac{\partial \pi_{\mathscr{D}}(\delta(\boldsymbol{\xi},\boldsymbol{\omega}_k))}{\partial \delta(\boldsymbol{\xi},\boldsymbol{\omega}_k)}\cdot\frac{\partial \delta(\boldsymbol{\xi},\boldsymbol{\omega}_k)}{\partial \boldsymbol{\omega}_l} \tag{53}
$$

as the derivative with respect to $\boldsymbol{\omega}_l$.

To combine the PLVQ with the CPN approach, we again assume $\boldsymbol{\xi}(\mathbf{x},W)$ to pe the sensoric response from the vector quantizer layer where the label is simply obtained $\mathbf{t}(\boldsymbol{\xi}) = \mathbf{t}(\mathbf{x})$ from the original data $\mathbf{x}$. Now, the cross-entropy (46) reads as

$$
Cr(\mathbf{t}(\mathbf{x})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}(\mathbf{x},W))) = \sum_c t_c(\mathbf{x})\cdot\log(p_{\mathcal{W}}(c|\boldsymbol{\xi}(\mathbf{x},W))) \tag{54}
$$

and we obtain

$$
\begin{aligned}
\frac{\partial Cr(\mathbf{t}(\mathbf{x})\,||\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi}(\mathbf{x},W)))}{\partial \mathbf{w}_l} &= \frac{\partial}{\partial \mathbf{w}_l}\left(\sum_c t_c(\mathbf{x})\cdot\log(p_{\mathcal{W}}(c|\boldsymbol{\xi}(\mathbf{x},W)))\right) \\
&= \sum_c \frac{t_c(\mathbf{x})}{p_{\mathcal{W}}(c|\boldsymbol{\xi}(\mathbf{x},W))}\cdot\frac{\partial p_{\mathcal{W}}(c|\boldsymbol{\xi}(\mathbf{x},W))}{\partial \mathbf{w}_l} \\
&\overset{(50)}{=} \sum_c \frac{t_c(\mathbf{x})}{p_{\mathcal{W}}(c|\boldsymbol{\xi}(\mathbf{x},W))}\cdot\frac{\partial}{\partial \mathbf{w}_l}\left(\sum_{j:c(\boldsymbol{\omega}_j)=c} S_{\mathcal{W}}(j,\boldsymbol{\xi}(\mathbf{x},W))\right) \\
&= \sum_c \frac{t_c(\mathbf{x})}{p_{\mathcal{W}}(c|\boldsymbol{\xi}(\mathbf{x},W))}\cdot\left(\sum_{j:c(\boldsymbol{\omega}_j)=c} \frac{\partial S_{\mathcal{W}}(j,\boldsymbol{\xi}(\mathbf{x},W))}{\partial \mathbf{w}_l}\right)
\end{aligned}
$$

as derivative of the cross-entropy with respect to the sensoric prototype $\mathbf{w}_l$. For the derivative $\frac{\partial S_{\mathcal{W}}(j,\boldsymbol{\xi}(\mathbf{x},W))}{\partial \mathbf{w}_l}$

we calulate

$$
\begin{aligned}
\frac{\partial S_{\mathcal{W}}\left(j,\boldsymbol{\xi}\left(\mathbf{x},W\right)\right)}{\partial \mathbf{w}_l} &= \frac{\partial}{\partial \mathbf{w}_l}\left(\frac{p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_j\right)\cdot p\left(\boldsymbol{\omega}_j\right)}{\sum_{k=1}^{N} p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)\cdot p\left(\boldsymbol{\omega}_k\right)}\right) \\
&= \frac{p\left(\boldsymbol{\omega}_j\right)\cdot\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_j\right)}{\partial \mathbf{w}_l}}{\sum_{k=1}^{N} p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)\cdot p\left(\boldsymbol{\omega}_k\right)} - p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_j\right)\cdot p\left(\boldsymbol{\omega}_j\right)\cdot\left(\frac{\sum_{k=1}^{N} p\left(\boldsymbol{\omega}_k\right)\cdot\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)}{\partial \mathbf{w}_l}}{\left(\sum_{k=1}^{N} p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)\cdot p\left(\boldsymbol{\omega}_k\right)\right)^2}\right) \\
&= \frac{p\left(\boldsymbol{\omega}_j\right)\cdot\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_j\right)}{\partial \mathbf{w}_l}}{\sum_{k=1}^{N} p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)\cdot p\left(\boldsymbol{\omega}_k\right)} - S_{\mathcal{W}}\left(j,\boldsymbol{\xi}\left(\mathbf{x},W\right)\right)\cdot\left(\frac{\sum_{k=1}^{N} p\left(\boldsymbol{\omega}_k\right)\cdot\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)}{\partial \mathbf{w}_l}}{\sum_{k=1}^{N} p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)\cdot p\left(\boldsymbol{\omega}_k\right)}\right) \\
&= \frac{p\left(\boldsymbol{\omega}_j\right)\cdot\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_j\right)}{\partial \mathbf{w}_l} - S_{\mathcal{W}}\left(j,\boldsymbol{\xi}\left(\mathbf{x},W\right)\right)\cdot\sum_{k=1}^{N} p\left(\boldsymbol{\omega}_k\right)\cdot\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)}{\partial \mathbf{w}_l}}{\sum_{k=1}^{N} p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)\cdot p\left(\boldsymbol{\omega}_k\right)}
\end{aligned}
$$

which includes the derivatives $\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_j\right)}{\partial \mathbf{w}_l}$. Using the dissimilarity density model (52) we have

$$
\frac{\partial p\left(\boldsymbol{\xi}\left(\mathbf{x},W\right)|\boldsymbol{\omega}_k\right)}{\partial \mathbf{w}_l} = \frac{\partial \pi_{\mathscr{D}}\left(\delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}_k\right)\right)}{\partial \delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}_k\right)}\cdot\frac{\partial \delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}_k\right)}{\partial \mathbf{w}_l} \tag{55}
$$

with the derivative $\frac{\partial \delta\left(\boldsymbol{\xi}\left(\mathbf{x},W\right),\boldsymbol{\omega}_k\right)}{\partial \mathbf{w}_l}$ according to (42) or to (43) for NG-based responses or fuzzy responses, respectively.

This PLVQ-approach (and its variants) can be summarized as

$$
X \underset{\text{NG/FCM}}{\leftrightarrows} W \underset{\text{NG-like/fuzzy}}{\overset{\boldsymbol{\xi}(\mathbf{x},W)}{\leftrightarrows}} \Xi \underset{\text{PLVQ}}{\overset{\mathbf{p}_{\mathcal{W}}(\boldsymbol{\xi})}{\longrightarrow}} \mathscr{C} \tag{56}
$$

in relation to (4).

# 4 Discussion

In this paper we describe formal extensions of counter propagation networks, which should make the original approach more flexible. In particular, we discussed several possibilities to transfer the data knowledge acquired by the vector quantization layer to the classification layer. Moreover, we considered several possibilities to replace the perceptron layer by alternative classification approaches. In the context of this we also studied how to realize a vector quantizer adaptation depending on the subsequent classification process realized in the second layer. Although this increases the complexity of the model, it could beneficial for certain applications.

At this point we did not discuss so far regularization aspects to achive model stability during learning and to control the flexibility. This should be done also in the context of the information bottleneck paradigm [47, 48] as well as reflecting the dilemma between information optimum data representation in the vector quantization layer and the demanded classification or regression performance [34, 36, 29].

An extension of the approach to the recently developed classification-by-components network (CbC, [41]) as alternative for LVQ variants should be investigated next.

# Acknowledgement

M. K. was supported by a ESF grant for a *Young Researcher Group*.

## References


[1] B. Bajželj and V. Drgan. Hepatotoxicity modeling using counter-propagation artificial neural networks: Handling an imbalanced classification problem. *Molecules*, 25(3):481, 2020.

[2] J.C. Bezdek. A convergence theorem for the fuzyy ISODATA clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(1):1–8, 1980.

[3] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum, New York, 1981.

[4] M. Biehl, B. Hammer, and T. Villmann. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, (2):92–111, 2016.

[5] Gail A. Carpenter and Stephen Grossberg. The ART of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer*, 21(3):77–88, 1988.

[6] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J.K. Su. This looks like that: Deep learning for interpretable image recognition. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 8930–8941. Curran Associates, Inc., 2019.

[7] M. Cottrell, J. C. Fort, and G. Pagès. Two or three things that we know about the Kohonen algorithm. Technical Report 31, Université Paris 1, Paris, France, 1994.

[8] K. Crammer, R. Gilad-Bachrach, A. Navot, and A.Tishby. Margin analysis of the LVQ algorithm. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing (Proc. NIPS 2002)*, volume 15, pages 462–469, Cambridge, MA, 2003. MIT Press.

[9] S. Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[10] Ed Erwin, Klaus Obermayer, and Klaus Schulten. Self-organizing maps: Ordering, convergence properties and energy functions. *Biol. Cyb.*, 67(1):47–55, 1992.

[11] T. Geweniger, F.-M. Schleif, and T. Villmann. Probabilistic prototype classification using t-norms. In T. Villmann, F.-M. Schleif, M. Kaden, and M. Lange, editors, *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of 10th International Workshop WSOM 2014, Mittweida*, volume 295 of *Advances in Intelligent Systems and Computing*, pages 99–108, Berlin, 2014. Springer.

[12] D. Graupe. *Principles of Artifiicial Neural Networks*, volume 8 of *Advanced Series in Circuits and Systems*, chapter Counter Propagation, pages 185–201. World Scientific, 3rd edition, 2019.

[13] S. Grossberg. Some networks that can learn, remember, and reproduce any number of complicated space-time patterns. *Journal of Mathematics and Mechanics*, 19(1):53–91, 1969.

[14] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[15] R. Hecht-Nielsen. Counter progagation networks. *Appl. Opt.*, 26(23):4979–4984, December 1987.

[16] R. Hecht-Nielsen. Review of 'self-organizing maps'. *IEEE Transactions on Neural Networks*, 7(6):1549–1550, November 1996.

[17] Robert Hecht-Nielsen. Applications of counterpropagation networks. *Neural Networks*, 1(2):131–139, 1988.

[18] John A. Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*, volume 1 of *Santa Fe Institute Studies in the Sciences of Complexity: Lecture Notes*. Addison-Wesley, Redwood City, CA, 1991.

[19] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–316. Elsevier, Amsterdam, 1999.

[20] X. Hou. Research on hyperspectral data classification based on quantum counter propagation neural network. *Advanced Materials Research*, 546–547:1377–1381, 2012.

[21] M. Kaden, M. Lange, D. Nebel, M. Riedel, T. Geweniger, and T. Villmann. Aspects in classification learning - Review of recent developments in Learning Vector Quantization. *Foundations of Computing and Decision Sciences*, 39(2):79–105, 2014.

[22] N. B. Karayiannis and J. C. Bezdek. An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering. *IEEE Transactions on Fuzzy Systems*, 5(4):622–8, 1997.

[23] Nicolaos B. Karayiannis and Pin I. Pai. Fuzzy algorithms for learning vector quantization. *IEEE Transactions on Neural Networks*, 7(5):1196–1211, 1996.

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

[25] Teuvo Kohonen. Learning Vector Quantization. *Neural Networks*, 1(Supplement 1):303, 1988.

[26] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[27] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(4):98–110, 1993.

[28] R. Krishnapuram and J. Keller. The possibilistic c-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems*, 4(3):385–393, 1996.

[29] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1294–1309, 2009.

[30] Thomas Martinetz and Klaus Schulten. Topology representing networks. *Neural Networks*, 7(2), 1994.

[31] Thomas M. Martinetz, Stanislav G. Berkovich, and Klaus J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[32] S. Musavishavazi, M. Kaden, and T. Villmann. Possibilistic classification learning based on contrastive loss in learning vector quantizer networks. In L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J.M. Zurada, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Soft Computing - ICAISC, Zakopane*, LNCS XXXXX, page in press, Cham, 2021. Springer International Publishing, Switzerland.

[33] D. Nebel, M. Kaden, A. Villmann, and T. Villmann. Types of (dis−)similarities and adaptive mixtures thereof for improved classification learning. *Neurocomputing*, 268:42–54, 2017.

[34] Karen L. Oehler and Robert M. Gray. Combining image compression and classification using vector quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:461–473, 1995.

[35] N.R. Pal, K. Pal, J.M. Keller, and J.C. Bezdek. A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems*, 13(4):517–530, 2005.

[36] Keren O. Perlmutter, Sharon M. Perlmutter, Robert M. Gray, Richard A. Olshen, and Karen L. Oehler. Bayes risk weighted vector quantization with posterior estimation for image compression and classification. *IEEE Trans. on Image Processing*, 5(2):347–360, February 1996.

[37] J.C. Principe. *Information Theoretic Learning*. Springer, Heidelberg, 2010.

[38] P. Ramachandran, B. Zoph, and Q.V. Le. Searching for activation functions. Technical Report arXiv:1710.05941v1, Google Brain, 2018.

[39] A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, 1961. University of California Press.

[40] Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wesley, Reading, MA, 1992.

[41] S. Saralajew, L. Holdijk, M. Rees, E. Asan, and T. Villmann. Classification-by-components: Probabilistic modeling of reasoning over a set of components. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 2788–2799. MIT Press, 2019.

[42] S. Saralajew, L. Holdijk, M. Rees, and T. Villmann. Robustness of generalized learning vector quantization models against adversarial attacks. In A. Vellido, K. Gibert, C. Angulo, and J.D.M. Guerrero, editors, *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization – Proceedings of the 13th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization, WSOM+2019, Barcelona*, volume 976 of *Advances in Intelligent Systems and Computing*, pages 189–199. Springer Berlin-Heidelberg, 2019.

[43] S. Saralajew, L. Holdijk, and T. Villmann. Fast adversarial robustness certification of nearest prototype classifiers for arbitrary seminorms. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, page in press. MIT Press, 2020.

[44] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[45] P. Schneider, B. Hammer, and M. Biehl. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[46] W. Sygnowski and B. Macukow. Counter-propagation neural network for image compression. *Optical Engineering*, 35(8):2214–17, 1996.

[47] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 1999.

[48] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. IEEE Information Theory Workshop (ITW), 2015. invited talk.

[49] A. Villmann, M. Kaden, S. Saralajew, and T. Villmann. Probabilistic learning vector quantization with cross-entropy for probabilistic class assignments in classification learning. In L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J.M. Zurada, editors, *Proceedings of the 17th International Conference on Artificial Intelligence and Soft Computing - ICAISC, Zakopane*, LNCS 10841, pages 736–749, Cham, 2018. Springer International Publishing, Switzerland.

[50] T. Villmann, A. Bohnsack, and M. Kaden. Can learning vector quantization be an alternative to SVM and deep learning? *Journal of Artificial Intelligence and Soft Computing Research*, 7(1):65–81, 2017.

[51] T. Villmann, J. Ravichandran, A. Villmann, D. Nebel, and M. Kaden. Investigation of activation functions for Generalized Learning Vector Quantization. In A. Vellido, K. Gibert, C. Angulo, and J.D.M. Guerrero, editors, *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization – Proceedings of the 13th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization, WSOM+2019, Barcelona*, volume 976 of *Advances in Intelligent Systems and Computing*, pages 179–188. Springer Berlin-Heidelberg, 2019.

[52] T. Villmann, S. Saralajew, A. Villmann, and M. Kaden. Learning vector quantization methods for interpretable classification learning and multilayer networks. In C. Sabourin, J.J. Merelo, A.L. Barranco, K. Madani, and K. Warwick, editors, *Proceedings of the 10th International Joint Conference on Computational Intelligence (IJCCI), Sevilla*, pages 15–21, Lissabon, Portugal, 2018. SCITEPRESS - Science and Technology Publications, Lda. ISBN: 978-989-758-327-8.

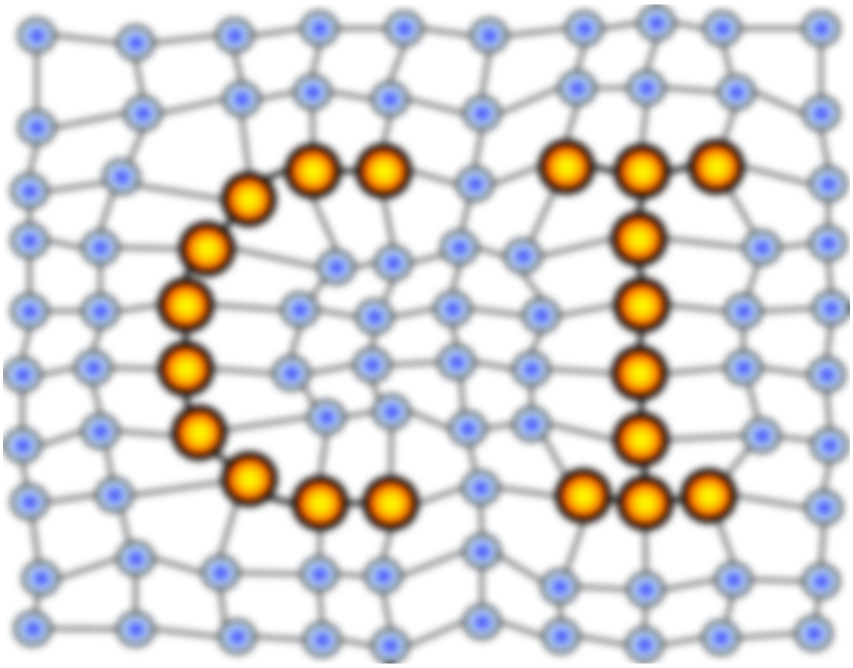
[53] M. Vracko. Kohonen artificial neural network and counter propagation neural network in molecular structure-toxicity studies. *Current Computer-Aided Drug Design*, 1(1):73–78, 2005.

[54] C. Wu, H.-L. Chen, and S.-C. Chen. Counter-propagation neural networks for molecular sequence classification: Supervised LVQ and dynamic node allocation. *Applied Intelligence*, 7:27–38, 1997.

[55] J. Zupan, M. Novic, and I. Ruisanchez. Kohonen and counterpropagation artificial neural networks in analytical chemistry. *Chemometrics and Intelligent Laboratory Systems*, 38:1–23, 1997.

# Machine Learning Reports

Report 01/2021



**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

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