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 Schubert¹, Marika Kaden ¹ and Thomas Villmann¹
(1) University of Applied Sciences Mittweida
Technikumplatz 17, 09648 Mittweida - Germany
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 robustness, 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 also 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 and 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 \( \mathcal{N} = \{1, \ldots, k\} \subset \mathbb{N}_+ \) equipped with weight vectors but here interpreted as prototypes (reference vectors) \( \mathcal{P} = \{p_1, \ldots, p_k\} \subset \mathbb{R}^n \) for data representation. To create a network response, we suppose a given data sample (stimulus) \( x \) from the data set \( \mathcal{X} \subset \mathbb{R}^n \). The response of the network to the given stimulus \( x \) obeys the winner-takes-all rule (WTA-rule):

\[
\nu(x, \mathcal{P}) = \arg \min_{j \in \mathcal{N}} d(x, p_j),
\]

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

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

is valid.

To adapt NG to the dataset $\mathcal{X}$, Martinetz et al. (1993) proposed an adaptation rule for the prototypes $\mathcal{P}$ 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 $x$ every prototype $p_j$ is updated according to

$$\Delta p_j = -\epsilon \cdot h_\lambda(x, p_j, \mathcal{P}) \cdot \frac{\partial d(x, p_j)}{\partial p_j} ,$$

(2)

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

$$h_\lambda(x, p_j, \mathcal{P}) = \exp \left( -\frac{r k(x, p_j, \mathcal{P})}{\lambda} \right) ,$$

(3)

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

$$r k(x, p_j, \mathcal{P}) = \sum_{l \in \mathcal{N}} H(d(x, p_j) - d(x, p_l)) ,$$

(4)

where $H(z)$ is the Heaviside step function, i.e. $H(z) = 1 \iff z > 0$ and 0 else holds. Considering $\mathcal{X}$ as a medium (gas) and $\lambda$ 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:

$$E_{\text{NG}} = E(\mathcal{X}, \mathcal{P}, \lambda) = \eta(\lambda) \sum_{j \in \mathcal{N}} \int_{x \in \mathcal{X}} P(x) \cdot h_\lambda(x, p_j, \mathcal{P}) \cdot (x - p_j)^2 dx ,$$

(5)

where $P(\mathcal{X})$ is the (unknown) data distribution and $\eta(\lambda)$ is a normalization constant depending on $\lambda$. 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 $f(x) = y_x \in \mathcal{Y}$, $\forall x \in \mathcal{X}$ 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 $\pi_j$ are only responsible for the data contained in the corresponding Voronoï-cell $V_j$ using
In this way, each neuron is equipped with an adaptive predictor \( \pi_j : \mathbb{R}^n \rightarrow \mathbb{R}^m \). 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

\[
\pi_j(x) = w_j^\top (x - p_j) + b_j,
\]

depending on the vector difference \( d_j = x - p_j \) and a weight vector \( w_j \in \mathbb{R}^n \) together with a bias \( b_j \in \mathbb{R} \) such that \( \pi_j : \mathbb{R}^n \rightarrow \mathbb{R} \forall x \in V_j \) is valid. Thus, the NG dynamic first determines the placement of the prototypes and afterwards the predictors are optimized by using the cost function

\[
E_{NGP} = \sum_{j \in N} \int_{x \in X} P(x) \cdot h_\lambda(x, p_j, P) \cdot (\pi_j(x) - y_x)^2 dx,
\]

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

### 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 \( C = \{1, ..., L\} \), each data sample \( x \) can be assigned to one of the \( L \) classes in an possibilistic manner, yielding \( c_x = (c_{x1}, ..., c_xL) \) 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 \( R(\pi_j) \subseteq [0, 1]^L \) is assumed. Thus, following the prediction steps of NGTSP for a stimulus \( x \), the response of FLNG is obtained as

\[
\pi_s(x) = c_{p_s},
\]

with \( s = \nu(x, P) \), which results in assigning the possibilistic class vector of the closest prototype \( p_s \) to stimulus \( x \).

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

\[
C_{FLNG} = \beta \cdot E_{NG} + (1 - \beta) \cdot E_{FL},
\]

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

\[
E_{FL} = E_{FLD} = \frac{1}{2} \sum_{j \in N} \sum_{x \in X} h_\lambda(x, p_j, P) \cdot (\pi_j(x) - c_x)^2,
\]

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

\[
E_{FL} = E_{FLC} = \frac{1}{2} \sum_{j \in N} \int_{x \in X} P(x) \cdot g_\gamma(x, p_j) \cdot (\pi_j(x) - c_x)^2 dx,
\]

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 $\beta$.

## 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 $d_j = (x - p_j)$. In particular,

1. for dense clusters in $\mathcal{X}$ it is likely that $d_j \to 0$, yielding:
   \[
   \pi_j(x) \xrightarrow{d_j \to 0} b_j, \tag{12}
   \]
   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 $x_1, x_2 \in V_j$ and $x_l = p_j + \alpha_l \cdot t$ for $l \in \{1, 2\}$, i.e. we can draw a line through the prototype $p_j$ connecting $x_1$ and $x_2$. However, due to the definition in (6) $d_j$ causes the prediction for such a case to be mainly compensated by the offset $b_j$. Thus, considering the prediction of $x_l$, we obtain
   \[
   \pi_j(x_l) = w_j^\top (x_l - p_j) + b_j \tag{13}
   \]
   \[
   = w_j^\top [(p_j + \alpha_l \cdot t) - p_j] + b_j \tag{14}
   \]
   \[
   = \alpha_l \cdot w_j^\top t + b_j. \tag{15}
   \]

Now, since $x_1, x_2$ are connected by a line going through $p_j$, we can consider the case $-\alpha_1 \approx \alpha_2$, yielding for the respective predictions
   \[
   \pi_j(x_1) \approx \alpha_1 \cdot w_j^\top t + b_j, \tag{16}
   \]
   and for $x_2$
   \[
   \pi_j(x_2) \approx -\alpha_2 \cdot w_j^\top t + b_j, \tag{17}
   \]
   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 definition in (6) for a regression model and define instead
\[ \tilde{\pi}_j(x) = w_j^\top \tilde{x} + b_j, \]  
(19)
where \( \tilde{x} \) can be the stimulus \( x \) itself or any transformation \( \tilde{x} = T(x) \) (e.g. a polynomial transformation) allowing also for non-linear regression architectures and thus for more flexibility.

Note, that in the trivial case \( \tilde{x} = x \), the symmetry consideration can still be applied, however expressing \( x = p_j + \alpha_x \cdot t \), yields
\[ \pi_j(x) = w_j^\top p_j + \alpha_x \cdot w_j^\top t + b_j, \]  
(20)
where we find that a differing sign is further compensated by \( p_j \) itself.

Furthermore, note that \( \tilde{d}_j = T(d_j) \) is also possible, however care needs to be taken when choosing \( T(\cdot) \), since \( d_j \to 0 \) 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 \textit{Regression Neural Gas} (RegNG)
\[ E_{\text{RegNG}} = \beta \cdot E_{\text{NG}} + (1 - \beta) \cdot E_{\text{Reg}}, \]  
(21)
with \( E_{\text{Reg}} \) as the regression error caused by the current prototype placement, which is defined as
\[ E_{\text{Reg}} = \sum_{j \in \mathcal{N}} \int P(x) \cdot g_y(x, p_j) \cdot (\tilde{\pi}_j(x) - y_x)^2 dx. \]  
(22)
Which can be interpreted as an supervised alternative to NGTSP.

Yet another approach is to additionally 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 \textit{regression-sensitive} ranking \( r_{k_{RS}} \). Considering an arbitrary \( \tilde{\pi}_r \) yields
\[ r_{k_{RS}}(x, \tilde{\pi}_r, \Pi) = \sum_{j \in \mathcal{N}} H(\rho_r(x) - \rho_j(x)), \]  
(23)
with
\[ \rho_j(x) = (\tilde{\pi}_j(x) - y_x)^2. \]  
(24)
Consequently, we replace the function \( h_\lambda(\cdot) \) in (3) by
\[ h_{RS}(\lambda, x, \tilde{\pi}_r, \Pi) = \exp \left( -\frac{r_{k_{RS}}(x, \tilde{\pi}_r, \Pi)}{\lambda} \right). \]  
(25)
We will refer to this model as \textit{Regression-Sensitive Neural Gas} (RegSeNG) describing the cost function
\[ E_{\text{RegSeNG}} = \beta \cdot E_{\text{RSNG}} + (1 - \beta) \cdot E_{\text{Reg}}, \]  
(26)
where \( E_{\text{RSNG}} \) 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


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


Machine Learning Reports

Report 01/2023

Impressum

Machine Learning Reports

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
Copyright of the articles remains to the authors.

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

ISSN: 1865-3960