ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Variants of Neural Gas for Regression Learning

Ronny Schubert, Marika Kaden and Thomas Villmann*

Mittweida University of Applied Sciences
Saxon Institute for Computational Intelligence and Machine Learning
Technikumplatz 17, 09648 Mittweida

Abstract. Approximation problems, and thus regression problems, have been widely considered as machine learning problems. A popular model to tackle such tasks are *radial-basis-function networks* (**RBFN**) and variants thereof. However, due to the global approximation scheme, RBFN, when trained in a supervised manner without additional constraints, may lack local representation. To this end, we propose approaches that aim to preserve locality in terms of the regression problem by using the Neural Gas algorithm. The models are tested on different data sets and compared to the supervised RBFN approach.

1 Introduction

During the last years, machine learning methods became a promising alternative to classical numerical methods for regression modeling. Thus, those approaches can be seen as comparatively sparse approximation approaches realized by rather small artificial neural networks [1], which constitute a network variant based on the *radial-basis-function network* (**RBFN**) proposed by [2]. Generally, RBFNs have become a popular tool for approximation-related tasks due to its sparsity and the fact that certain RBFs are universal approximators [1]. However, to achieve this sparsity, RBFN requires RBF-centers that are similar to prototypes from *vector quantization* (**VQ**). Therefore, VQ methods generally are being considered to initialize or adapt these centers. Accordingly, prototype-based regression models inherit the properties known from VQ methods and its variations. Main advantages given by VQ are interpretability and robustness [3, 4, 5], whereas black box models like *multi-layer-perceptrons* (**MLP**) are in need of explainability. Yet, techniques to gain insight into a black box model in terms of regression were recently proposed by [6] in which the *restructuring* approach involves a reference point, which can in this regard be interpreted as a prototype in the linear layer fulfilling certain rules. Nonetheless, when trained in supervised (backpropagation) manner, i.e. with all necessary parameters adjusted, RBFNs tend to produce questionable prototype placements without additional constraints on representation [2, 7], yielding a lack of interpretability and transperancy. In this regard, we shall propose two models which aim to keep an interpretable representation and, thus, locality. The respective models are an supervised extension of the *hybrid* approach of *Neural Gas* (**NG**) which was also used for function approximation [8]. In this work, we shall refer to two approaches - the *hybrid* approach, which uses VQ as initialization and does not adapt the prototypes based on the regression task, and *supervised* in which the

*R. S. is supported by grants of the Agriculture Ministry of Germany (28DK112A20).

placement is additionally influenced by the regression task and corresponding parameters are accordingly adapted.

2 The Regression Problem

To clarify the notations we briefly describe the regression problem. For this we assume, that we are given a training set $\mathcal{T} = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$ with $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^n$ and $y_i \in \mathcal{Y} \subseteq \mathbb{R}$. The regression task can then be formulated as

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i \tag{1}$$

such that a target y_j is supposed to be described by the function $f(\boldsymbol{x}_j)$ and some additive noise ϵ_i. However, in general little to nothing is known about $f(\cdot)$, and thus the goal is to approximate this function using the information given by the data $\mathcal{X}$ and the corresponding targets $\mathcal{Y}$.

3 Neural Gas and its application in regression

We decided for NG as VQ model, since the approximation performance is known to be superior compared to k-means and *self-organizing-maps* [8]. Here we will use an unnormalized version of the original cost function given by [8]

$$C_{NG} = \sum_k \int_{\boldsymbol{x} \in \mathcal{X}} h_\lambda(\boldsymbol{x}, \boldsymbol{p}_k, \mathcal{P}) d(\boldsymbol{x}, \boldsymbol{p}_k) \tag{2}$$

for a prototype set $\mathcal{P} \subset \mathbb{R}^n$, the distance measure $d(\cdot)$, which is usually chosen as the squared euclidean distance, and the neighborhood cooperativeness

$$h_\lambda(\boldsymbol{x}, \boldsymbol{p}_j, \mathcal{P}) = exp\left(-\frac{rk(\boldsymbol{x}, \boldsymbol{p}_j, \mathcal{P})}{\lambda(t)}\right) \quad \text{with} \quad \lambda(t) \xrightarrow{t \to \infty} 0 \tag{3}$$

with $rk(\cdot)$ as a counting of the condition $d(\boldsymbol{x}, \boldsymbol{p}_j) \leq d(\boldsymbol{x}, \boldsymbol{p}_i)$ being satisfied. Additionally, $\lambda(t)$ models the visibility dependend on training time t. The authors in [8] also applied NG to an approximation related problem - the prediction of time-series data, which we will abreviate for distinction as **NGTSP** (*Neural Gas for time-series prediction*). However, as already mentioned, the proposed scheme is in a hybrid mode in which NG is used for the placement of the prototypes and thereafter the following regression-based discretized cost function is optimized:

$$C_{NGTSP} = \sum_k \sum_i h_{\hat{\lambda}}(\boldsymbol{x}_i, \boldsymbol{p}_k, \mathcal{P})(y_i - \pi_k(\boldsymbol{x}_i))^2 \tag{4}$$

with

$$\pi_j(\boldsymbol{x}) = \boldsymbol{w}_j^\intercal \cdot (\boldsymbol{x} - \boldsymbol{p}_j) + \beta_j \tag{5}$$

such that (4) realizes an optimization of the mean squared error of the prediction and (1) is approximated by a partition of $f(\cdot)$ via assigning each data sample to its winning prototype, i.e. prediction is done locally through the *winner-takes-all* (**WTA**) scheme

$$f(\hat{\boldsymbol{x}}) = \pi_i(\hat{\boldsymbol{x}}) \iff i = \arg\min_j d(\hat{\boldsymbol{x}}, \boldsymbol{p}_j) \tag{6}$$

which is in contrast to the global approach of RBFN, where every prototype participates in the prediction. Further, the schedule for $\hat{\lambda}(t)$ in (5) does not necessarily need to match the one of $\lambda(t)$.

4 Extending NGTSP to a supervised scenario

To extend NGTSP from a hybrid into a supervised setting we follow the ideas of [9] for an supervised extension of NG in terms of fuzzy-labeling and the considerations of [10] for combining representation and supervised tasks. In consequence, we need to balance between data representation and placing prototypes such that prediction is optimized. We propose the following cost function

$$C_{RegNG} = \alpha C_{NG} + (1-\alpha) C_{Reg} \tag{7}$$

and denote the corresponding model *Regression Neural Gas* (**RegNG**). Due to space constraints, we refer the reader to [11] for a comprehensive description. In (7) $\alpha \in [0,1]$ is a parameter balancing representation costs by usual NG and the regression costs

$$C_{Reg} = \sum_k \sum_i g_{\hat{\lambda}}(\boldsymbol{x}_i, \boldsymbol{p}_k)(y_i - \tilde{\pi}_k(\boldsymbol{x}_i))^2 \tag{8}$$

where $g_{\hat{\lambda}}(\cdot)$ is chosen RBF-like as described in [9] depending on the visibility $\hat{\lambda}(t)$. The predictor $\tilde{\pi}_j(\boldsymbol{x})$ in (8) is defined as

$$\tilde{\pi}_j(\boldsymbol{x}) = \boldsymbol{w}_j^{\intercal} \cdot T(\boldsymbol{x}) + \beta_j \tag{9}$$

in which we allow $T(\boldsymbol{x})$ to be a transformation on $\boldsymbol{x}$ resulting in greater flexibility for the approximation. For example, $T(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{p})$ would correspond to (5), whereas $T(\boldsymbol{x})$ taken as a polynomial transform of $\boldsymbol{x}$ yields a non-linear approximation. Prediction is then done in the same manner as in (6). Yet, another way to incorporate the information of the regression can be done through the neighborhood cooperativeness (3). In this regard, we alter the counting to not be dependent on the distances, but rather on their corresponding predictors, i.e. the counting is now sensitive to the condition $(\tilde{\pi}_j(\boldsymbol{x}_i) - y_i)^2 \leq (\tilde{\pi}_k(\boldsymbol{x}_i) - y_i)^2$ being satisfied. The resulting model is called *regression-sensitive Neural Gas* (**RegSeNG**) with the cost function

$$C_{RegSeNG} = \alpha C_{RSNG} + (1-\alpha) C_{Reg} \tag{10}$$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

4.1 Relations between Learning Vector Quantization, supervised RBFN and RegNG

Since the goal is to establish a VQ-based regression model using gradient techniques, we may find parallels of *learning vector quantization* (**LVQ**), RegNG and supervised RBFN (**SRBFN**). We consider the general expression of RBFN [1]

$$f(\boldsymbol{x}) = \sum_k c_k G(\boldsymbol{x} - \boldsymbol{p}_k) \tag{11}$$

with $G(\cdot)$ being an arbitrary RBF centered at $\boldsymbol{p}_k$. When trained in supervised gradient descent manner, the corresponding prototype or center updates are influenced by their respective weights, i.e. for prototype $\boldsymbol{p}_j$ the weight c_j. This, together with the fact, that c_j is allowed to be negative, can be related to LVQ schemes: LVQ models attract and repell prototypes based on certain rules, i.e. attraction (repulsion) is realized through a match (mismatch) of classes, which is expressed as positive (negative) sign [12]. An regression-based LVQ-variant can be found in [13] which is called *regression-LVQ* (**RLVQ**). Hence, in this regard, SRBFN consists of an attraction and repulsion scheme, but in terms of regression and it can further be extended, showing that RBFN in special cases are reducible to LVQ schemes [14]. However, when considering RegNG (RegSeNG) we find that C_{NG} (C_{RSNG}) realizes the attraction, while C_{Reg} (8) corresponds to repulsion. This can be verified by simple mathematical calculations, which are dropped here due to the lack of space.

It should be emphasized that RegNG (RegSeNG) aims to place prototypes in such a way that target trends are found with respect to the chosen transformation $T(\boldsymbol{x})$. For example, assuming $T(\boldsymbol{x})$ is linear, then the repulsion depends locally only on the linearity of the targets within the responsibility area of the prototypes. Nonetheless LVQ, SRBFN and RegNG might encounter representation problems [15, 7]. However, considering (7) we find that the balancing is crucial to control representation and, if not chosen correspondingly, local optimization might suffer.

5 Experiments[1]

For the experiments we have chosen the datasets *California Housing*[2] (**CH**), *Breast Cancer Prognostic (wpbc)*[3] (**BC**), *Diabetes*[4] (**DB**) and *Wine Quality - Red*[5] (**WQ**). In WQ we picked *alcohol* as the target and for BC we removed the attributes ID, Outcome and Lymph Node status and for the target we took the mean perimeter. All datasets were normalized, such that $\mathcal{X} \subseteq [0,1]^n$ and $\mathcal{Y} \subseteq [0,1]$. No further feature extraction was applied. We ran each model with 5, 10 and 15 prototypes, respectively, and used an 5-fold cross validation. As

[1] A comprehensive overview of the experimental setting and results: `https://github.com/rmschubert/RegressionVQ`

[2] `https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html`

[3] `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Prognostic)`

[4] `https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html`

[5] `https://archive.ics.uci.edu/ml/datasets/Wine+Quality`

evaluation measures we decided for the average of the coefficent of determination r^2 and the standard error *sep* on the test sets. We compared our approaches to the SRBFN with a gaussian-like RBF, in which we equipped each center with an own parameter γ to determine the scaling and adapted both centers and scaling. The other models for comparison are the *soft*-RLVQ [13], the NGTSP and a variant of it in which we simply replace $(\boldsymbol{x} - \boldsymbol{p})$ in (5) by $\boldsymbol{x}$ which we call for distinction **xNGTSP**. Futhermore, for Reg(Se)NG we used linear regression, such that $T(\boldsymbol{x}) = \boldsymbol{x}$. All models were updated by using **ADAM**[6] as gradient descent technique.

Table 1: Average performance of the models per number of prototypes $|\mathcal{P}|$ and Dataset

DS	$\lvert\mathcal{P}\rvert$	**SRBFN**		**RegNG**		**RegSeNG**	
		r^2	sep	r^2	sep	r^2	sep
	5	0.97 ± 0.02	0.03 ± 0.01	0.94 ± 0.03	0.04 ± 0.01	0.90 ± 0.06	0.05 ± 0.02
BC	10	0.97 ±0.02	0.03 ±0.01	0.89 ±0.12	0.05 ±0.02	0.94 ±0.02	0.04 ±0.01
	15	0.98 ±0.01	0.02 ±0.01	0.95 ±0.03	0.03 ±0.01	0.87 ±0.17	0.05 ±0.03
	5	0.64 ± 0.01	0.01 ± 0.00	0.39 ± 0.25	0.01 ± 0.00	0.39 ± 0.30	0.01 ± 0.00
CH	10	0.65 ±0.02	0.01 ±0.00	0.36 ±0.28	0.01 ±0.00	0.36 ±0.28	0.01 ±0.00
	15	0.65 ±0.02	0.01 ±0.00	0.40 ±0.23	0.01 ±0.00	0.37 ±0.26	0.01 ±0.00
	5	0.51 ± 0.06	0.10 ± 0.02	0.51 ± 0.05	0.10 ± 0.01	0.50 ± 0.06	0.11 ± 0.01
DB	10	0.51 ±0.06	0.10 ±0.01	0.49 ±0.06	0.11 ±0.01	0.50 ±0.06	0.11 ±0.01
	15	0.51 ±0.06	0.10 ±0.01	0.50 ±0.06	0.10 ±0.01	0.51 ±0.06	0.11 ±0.01
	5	0.72 ± 0.04	0.04 ± 0.00	0.67 ± 0.04	0.04 ± 0.00	0.67 ± 0.05	0.04 ± 0.00
WQ	10	0.74 ±0.05	0.03 ±0.00	0.68 ±0.03	0.04 ±0.00	0.67 ±0.03	0.04 ±0.00
	15	0.74 ±0.05	0.03 ±0.00	0.69 ±0.04	0.04 ±0.00	0.68 ±0.04	0.04 ±0.00

DS	$\lvert\mathcal{P}\rvert$	**RLVQ**		**NGTSP**		**xNGTSP**	
		r^2	sep	r^2	sep	r^2	sep
	5	0.90 ± 0.02	0.74 ± 0.21	0.87 ± 0.04	0.07 ± 0.01	0.82 ± 0.08	0.07 ± 0.02
BC	10	0.90 ±0.04	0.67 ±0.12	0.84 ±0.06	0.06 ±0.01	0.85 ±0.02	0.06 ±0.01
	15	0.91 ±0.03	0.67 ±0.13	0.75 ±0.22	0.08 ±0.04	0.68 ±0.17	0.09 ±0.02
	5	0.51 ± 0.03	0.96 ± 0.08	0.33 ± 0.12	0.04 ± 0.01	0.42 ± 0.29	0.01 ± 0.00
CH	10	0.51 ±0.04	0.87 ±0.10	0.57 ±0.03	0.01 ±0.00	0.40 ±0.27	0.01 ±0.00
	15	0.52 ±0.03	0.93 ±0.01	0.61 ±0.02	0.01 ±0.00	0.43 ±0.25	0.01 ±0.00
	5	0.42 ± 0.07	1.97 ± 0.43	0.41 ± 0.08	0.13 ± 0.01	0.51 ± 0.05	0.10 ± 0.01
DB	10	0.43 ±0.08	1.84 ±0.43	0.49 ±0.07	0.11 ±0.01	0.48 ±0.06	0.10 ±0.02
	15	0.43 ±0.08	1.80 ±0.14	0.46 ±0.06	0.11 ±0.01	0.48 ±0.06	0.10 ±0.01
	5	0.43 ± 0.07	3.25 ± 0.89	0.46 ± 0.06	0.06 ± 0.00	0.69 ± 0.07	0.04 ± 0.01
WQ	10	0.46 ±0.07	3.75 ±0.59	0.51 ±0.07	0.06 ±0.01	0.72 ±0.04	0.03 ±0.00
	15	0.44 ±0.08	3.69 ±1.13	0.56 ±0.04	0.05 ±0.01	0.71 ±0.04	0.04 ±0.00

In Table 1 the results of the experiments are shown. We find that in some cases it can already be sufficient to use a hybrid NG scheme, as the results for NGTSP in the case of CH and xNGTSP for WQ are indicating. However, we find that Reg(Se)NG are on par with SRBFN for DB and BC and at least close for WQ. Nonetheless, both models show high deviations in certain cases for r^2, reaching in peak performance comparable results to SRBFN. These deviations can be due to an inappropriate balancing and the constraint of the linear approximation, while additionally finding local trends, which must not transfer to unseen data. This gives rise to varying definitions of $T(\boldsymbol{x})$. Additionally, we encountered states for *every* supervised model in which some prototypes were not representive. However, in Reg(Se)NG this can be circumvented with a dataset specific balancing, yielding direct control over the optimization and representation, while SRBFN need to be additionally modified [7].

[6] https://arxiv.org/abs/1412.6980

6 Summary and Outlook

We could show that a local approach to combine VQ and supervised (linear) regression could keep up with the performance of SRBFN in certain cases. However, further investigations can be concerned with different definitions of $T(\boldsymbol{x})$ or prototype pruning and adding to reduce the dependency on the balancing, as it was proposed for (S)RBFN as well [1]. This can easily be integrated due to the WTA-rule (6) for prediction. As it was contemplated by [16] a weighted norm instead of the squared euclidean norm could be used. Together with the regression-based representation, this can be considered as relevance learning in VQ [17].

References

[1] T. Poggio and F. Girosi. A theory of networks for approximation and learning. Technical report, 1989.

[2] J. Moody and Ch. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, jun 1989.

[3] O. Li, H. Liu, C. Chen, and C. Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *The Thirty-Second AAAI Conferenceon Artificial Intelligence (AAAI-18)*, number 432, pages 3530–3537, 2018.

[4] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Survey*, 16:1–85, 2022.

[5] P. Lisboa, S. Saralajew, A. Vellido, R. Fernández-Domenech, and T. Villmann. The coming of age of interpretable and explainable machine learning models. *Neurocomputing*, 535:25–39, 2023.

[6] S. Letzgus, P. Wagner, Jo. Lederer, W. Samek, K.-R. Müller, and G. Montavon. Toward explainable artificial intelligence for regression models: A methodological perspective. *IEEE Signal Processing Magazine*, 39(4):40–58, 2022.

[7] L. J. Herrera, H. Pomares, I. Rojas, A. Guillén, G. Rubio, and J. Urquiza. Global and local modelling in radial basis functions networks. In *Bio-Inspired Systems: Computational and Ambient Intelligence*, pages 49–56. Springer Berlin Heidelberg, 2009.

[8] T.M. Martinetz, S.G. Berkovich, and K.J. Schulten. 'neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, jul 1993.

[9] Th. Villmann, B. Hammer, F. Schleif, T. Geweniger, and W. Herrmann. Fuzzy classification by fuzzy labeled neural gas. *Neural Networks*, 19(6-7):772–779, jul 2006.

[10] K.L. Oehler and R.M. Gray. Combining image compression and classification using vector quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):461–473, 1995.

[11] R. Schubert, M. Kaden, and T. Villmann. Regression neural gas: Extension of standard neural gas and its application for function approximation. In *Machine Learning Reports 01/2023*, 2023.

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

[13] M. Grbovic and S. Vucetic. Regression learning vector quantization. In *2009 Ninth IEEE International Conference on Data Mining*. IEEE, dec 2009.

[14] P. Frasconi, M. Gori, and G. Soda. Links between lvq and backpropagation. *Pattern Recognition Letters*, 18(4):303–310, 1997.

[15] Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.

[16] Federico Girosi. Regularization theory, radial basis functions and networks. In Vladimir Cherkassky, Jerome H. Friedman, and Harry Wechsler, editors, *From Statistics to Neural Networks*, pages 166–187, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[17] Petra Schneider, Michael Biehl, and Barbara Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, 2009.