Les Cahiers du GERAD ISSN: 0711–2440

Shallow Structured Potts Neural Network Regression (S-SPNNR)

K.-A. Alahassa Nonvikan
A. Murua

G–2020–23–EIW03

April 2020

Citation suggérée : K.-A. Alahassa Nonvikan, A. Murua (Avril 2020). Shallow Structured Potts Neural Network Regression (S-SPNNR), *In* C. Audet, S. Le Digabel, A. Lodi, D. Orban and V. Partovi Nia, (Eds.). Proceedings of the Edge Intelligence Workshop 2020, Montréal, Canada, 2–3 Mars, 2020, pages 15–20. Les Cahiers du GERAD G–2020–23, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2020-23-EIW03) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: K.-A. Alahassa Nonvikan, A. Murua (April 2020). Shallow Structured Potts Neural Network Regression (S-SPNNR), *In* C. Audet, S. Le Digabel, A. Lodi, D. Orban and V. Partovi Nia, (Eds.). Proceedings of the Edge Intelligence Workshop 2020, Montreal, Canada, March 2–3, 2020, pages 15–20. Les Cahiers du GERAD G–2020–23, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2020-23-EIW03) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Shallow Structured Potts Neural Network Regression (S-SPNNR)

Karl-Augustt Alahassa Nonvikan
Alejandro Murua

Département de mathématiques et de statistique, Université de Montréal, Montréal (Québec), Canada, H3T 1J4

alahassan@dms.umontreal.ca

April 2020
Les Cahiers du GERAD
G–2020–23–EIW03

Abstract: *We introduce a novel ensemble learning approach which combines random partitions models through Potts clustering with a non-parametric predictor such as shallow feedforward neural networks (S-SPNNR). Neural network are known as universal approximators, and are very well suited to explore others learning methods. We combine them with Potts clustering models to create a bagging-like learning framework where several estimates from each random partition are aggregated into one prediction. Our approach carries out the balance between overfitting and model stability in presence of small datasets with high dimensional features. We found that S-SPNNR is really effective in* multivariate multiple regression *task and present more predictive power than Multi-layer feedforward neural network and the Multi-layer Multi-target Regression (MMR) model given some datasets from the Mulan Multi-label learning project.*

1 Introduction

The model called *Structured Potts Neural Network* is an hierarchical Bayesian model where we train individual neural nets to specialize on sub-groups (latent clusters components) while we still stay informed about representations of the overall data. Our Potts neural network model differ from those of [1] and [4], which is a generalization of the Ising neural network. We call it a structured one, because we integrate the structured correlations among the weights (and offsets) of the network [5] through Markov Random Fields (MRF) process. Bayesian learning allows the opportunity to quantify posterior uncertainty on neural networks (NNs) model parameters. We can specify priors to inform and constrain our models and get structured uncertainty estimation.

The proposal is organized as follows. Section 2 presents the background framework, section 3 explains and presents the model as well as its three variations: the Shallow- Structured Potts Neural Network Regression (S-SPNNR) with Sparse Markov Random Fields (ShallowSparse), the S-SPNNR with fully Connected Markov Random Fields (ShallowFull), and the S-SPNNR with compound symetry matrix (ShallowSym). Section 4 and 5 show our results and present our concluding remarks respectively.

2 Background

2.1 Potts clustering

We present Potts Clustering based on [3] paper framework. The training data consists of n examples in the form of inputs vector $x = x_i \in \mathbb{R}^q$, and corresponding outputs $y = y_i$, where $y_i \in \mathbb{R}^{l_2}$ (a vector response) for each $i = 1, ..., n$. For our model, $x = x_i$ is the vector of available covariates for observation i.

As in [3], we assume a random partition model with a hierarchical form for these data :

$$y_1, ..., y_n | \rho_n, \psi_1^*, ..., \psi_{k_n}^* \overset{\text{ind}}{\sim} p(y_i | x_i, \psi_{s_i}^*) \tag{1}$$

$$\psi_1^*, ..., \psi_{k_n}^* \overset{\text{ind}}{\sim} p(\psi) \tag{2}$$

$$\rho_n \sim p(\rho_n | x) \tag{3}$$

where ρ_n is a partition of $[n]$ into k_n subsets, $s_1, ..., s_n$ are cluster membership indicators such that $s_i = j$ if the ith individual belongs to the jth cluster, and $\psi_i = \psi_{s_i}^*$ represent the *neural network* parameters for all $i \in [n]$.

Potts clustering model can be seen as a stochastic version of the label propagation approach [6]. In following section, we present the *feed-forward* network function itself, which is of the form $y = g(x, w, b)$, with w weights matrix, b biases matrix (offsets), and g an activation function.

2.2 The feed-forward neural network regression framework

The network itself is (in general) a multi-layer network, defined typically by the following equations. Layer k computes an output vector h^k using the output h^{k-1} of the previous layer, starting with the input $x = h^0$.

$$h^k = b^k \oplus g_k(h^{k-1})w^k \tag{4}$$

with parameter b^k (a vector of offsets/biases), w^k a matrix of weights, $\oplus$ the Kronecker sum, and g_k which is applied element-wise, represents any suitable non-linear function.

The top layer output h^l is used for making a prediction and is combined with the supervised target y into a loss function $L(h^l, y)$. The model output y is given by :

$$\mathbb{E}[y|h^{l-1}] = b^l \oplus h^{l-1}w^l$$

In what follows, a 2-layer network means that we build two (2) layer on top of the input layer.

3 The models

3.1 The S-SPNNR model with Sparse Markov Random Fields (ShallowSparse)

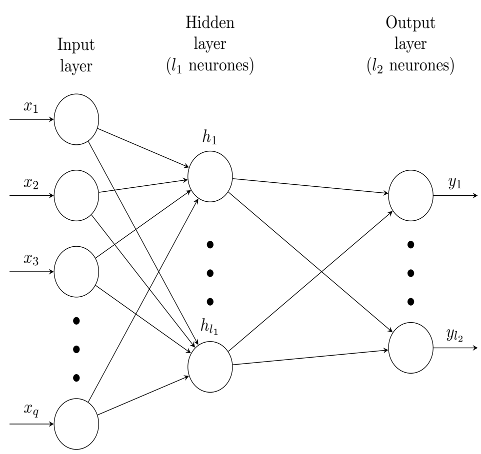

Figure 1: Shallow feedforward neural network

Given a Potts partition $\rho_n = (S_1, ..., S_{k_n})$ with k_n subsets, we denote by $\{\psi_1, \psi_2, ..., \psi_n\}$ the set of unique cluster-specific parameters. $y_j^* = \{y_i, i \in S_j\}$ and $x_j^* = \{x_i, i \in S_j\}$ denote respectively the set of responses and covariates of cluster S_j. Defining $h_i^2 = f_{\psi_j}(x_i)$, $h_{2j}^* = \{h_i^2, i \in S_j\}$.

$$p(y_j^*|h_{2j}^*, \psi_j, \Sigma) = \prod_{i \in S_j} (2\pi)^{-l_2/2}|\Sigma|^{-1/2} \times \exp\{-(1/2)(y_i - h_i^2)'\Sigma^{-1}(y_i - h_i^2)\} \tag{5}$$

with $\psi = (w^1, w^2, b^1, b^2)$ for each cluster. Our distribution specification for each y_i, $i = 1, ..., n$ is as follows:

$$\begin{aligned} y_i|x_i, \psi, \Sigma &\sim \mathcal{N}_{l_2}(f_\psi(x_i), \Sigma) \qquad (6) \\ p(y_i|x_i, \psi, \Sigma) &= (2\pi)^{-l_2/2}|\Sigma|^{-1/2} \exp\{-(1/2)[y_i - f_\psi(x_i)]'\Sigma^{-1}[y_i - f_\psi(x_i)]\} \end{aligned}$$

The architecture in each cluster is a 2-layers network. The model weights uncertainty is similarly measured as in [5] paper. As [9] and [10] have introduced a deep-structured conditional random field model which consists of multiple layers of simple Conditional Random Fields (CRFs) where each layer's input consists of the previous layer's input and the resulting marginal probabilities. We use the Markov Random Fields (MRFs) to set alike structure on the neural network weights and biases.

The weights MG-MRF is sparse and defined on vector $w = (\mathrm{vec}(w^1)^T, \mathrm{vec}(w^2)^T)$, with mean $\mu_= (\mu_1^T, \mu_2^T)$ (let's say $\mu_k = \mathbb{E}[\mathrm{vec}(w^k)]$), sparse precision matrix $\mathcal{J}$. For sparsity, we set only w_j^1 and w_i^2 as neighbors with $i = j$, where w_j^1 denotes the $j-$th column of w^1, and w_i^2 the $i-$th line of w^2.

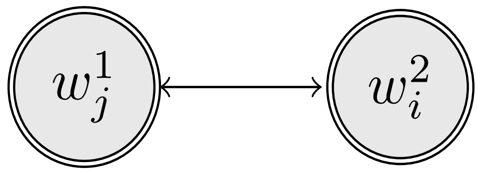

Figure 2: Sparse Multivariate Gaussian Markov Random Fields (MG-MRF) on the network weights

3.2 The S-SPNNR model with Fully Connected Markov Random Fields (ShallowFull)

The Fully Connected Markov Random Fields model is the same as described above with huge difference in weights connections. We set the whole matrices w^1 and w^2 as neighbors.

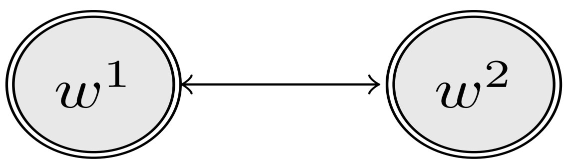

Figure 3: Fully Connected Markov Random Fields (MG-MRF) on the network weights

Fully-connected graphical models address issues of locally-connected models by assuming full connectivity amongst all nodes in the weights graph, thus taking full advantage of long range relationships to improve inference accuracy[8]. Just as importantly, in contrast to common fully-connected deep networks, we have less parameters in our case, thanks to the shallow network that present less connected layers.

3.3 The S-SPNNR-FCMRF model with compound symmetry matrix block (ShallowSym)

We have built for the Fully Connected Markov Random Field S-SPNNR model a compound symmetry version (ShallowSym) using the precision matrix $\mathcal{J}$. The matrix block $\mathcal{J}_{ii}$ for (w^1, w^2) itself can be express as a Kronecker product between two matrices U_i and V_i.

$$\mathcal{J}_{ii} = V_i \otimes U_i, U_i \in \mathcal{M}_{l_{i-1} \times l_{i-1}}, V_i \in \mathcal{M}_{l_i \times l_i}$$

To reduce the model complexity, we choose U_i and V_i to be a positive-definite matrix with compound symmetry structure (*constant diagonal and constant off-diagonal elements*). It means for example :

$$U_i = a_u I + (1 - \rho_u) \mathbb{1}\mathbb{1}^T$$

where a_u is a strictly positive number, and ρ_u a real-number. I is an identidy matrix with dimension l_{i-1}, and $\mathbb{1}$ a vector of ones of size l_{i-1}. In a more interpretive manner, a_u represent the intra-class correlation accross the weights and $a_u + (1 - \rho_u)$ their total variance [2] in the case V_i is estimated as a matrix of ones. This configuration is more likely usefull when all the variances may be nearly equal, and the covariances may be nearly equal among all the scalar weights at each layer. Those constraints save a lot of degrees of freedom with little loss of fit, because we only have to estimate one variance and one covariance for U_i.

4 Experimental evaluation

4.1 Datasets

The performance of the S-SPNNR in his three versions (ShallowSparse, ShallowFull and ShallowSym) were experimentally evaluated. The Mulan project [7] was used to evaluate the results. The experiments were performed on 11 multi-output regression datasets (see Table 1 below) that are among the benchmark data available from the Mulan project website.[1]

Table 1: Summary of data sets characteristics: name, domain, number of instances, features and targets

Data sets	Domain	Instances	Numb. of attributes	Numb. of targets
Andromeda	Water	49	30	6
Slump	Concrete	103	7	3
EDM	Machining	154	16	2
ATP7D	Forecast	296	211	6
ATP1D	Forecast	337	411	6
Jura	Geology	359	15	3
Online sales	Forecast	639	401	12
ENB	Buildings	768	8	2
Water quality	Biology	1 060	14	16
SCPF	Forecast	1 137	23	3
River flow 1	Forecast	9 125	64	8

We have also compared the performance of our models against the Multi-layer Multi-target Regression (MMR) model [11] that haved already substantially outperformed the best results from state-of-the-art algorithms on most of those 11 datasets and a 5-layer feedforward regression network (5-layer FFRNN).

To directly benchmark with state-of-the-art algorithms, we measure the performance by the commonly-used Relative Root Mean Squared Error (RRMSE) defined as :

$$\sqrt{\frac{\sum_{(xi,yi)\in Dtest}(\hat{y}_i - y_i)^2}{\sum_{(xi,yi)\in Dtest}(\hat{Y} - y_i)^2}}$$

where (x_i, y_i) is the i-th sample x_i with ground truth target y_i, $\hat{y}_i$ is the prediction of y_i and $\hat{Y}$ is the average of the targets over the training set $Dtrain$. We take the average RRMSE (aRRMSE) across all the target variables within the test set $Dtest$ as a single measurement. It measures the root squared error relative to what it would have been if a simple predictor had been used. A lower aRRMSE indicates better performance.

5 The results

Compare to a simple predictor, the proposed S-SPNNR model and its three versions have achieved great results against the MMR model. This large improvement of the proposed S-SPNNR over the MMR with significant margins on all the 11 datasets shows its effectiveness modeling multi-target regression task. Andromeda, and SCPF show that the 5-layer FFRNN is still beatable in terms of predictive power for these datasets. ShallowSparse was really effective on EDM, ATP7D, Jura, online sales and water quality against the ShallowFull and ShallowSym. ShallowSim was better against ShallowFull only on slump, ENB and water quality.

[1] http://mulan.sourceforge.net/datasets-mtr.html

Table 2: Summary of aRRMSE (%) obtained with S-SPNNR and MMR models

Data sets	‘ MMR	ShallowFull	ShallowSym
Andromeda	52.7	31.63	32.35
Slump	58.7	21.90	18.47
EDM	71.6	28.01	35.96
ATP7D*	44.3	22.69	24.56
ATP1D*	33.2	13.50	14.63
Jura	58.2	28.98	25.81
Online sales*	70.9	18.90	21.59
ENB*	11.1	39.05	45.79
Water quality	88.9	10.01	8.26
SCPF	81.2	12.30	13.86
River flow 1*	8.9	10.97	11.45

* *We reduce the input features to the first 6 PCA components.*

Table 3: Summary of aRRMSE (%) obtained with S-SPNNR and the 5-layer FFRNN model

Data sets	‘ ShallowSparse	5-layer FFRNN
Andromeda	30.91	37.44
Slump	20.02	19.83
EDM	17.23	15.71
ATP7D*	19.73	13.67
ATP1D*	29.54	9.89
Jura	13.46	8.15
Online sales*	14.79	8.78
ENB*	23.92	4.36
Water quality	6.48	6.15
SCPF	10.78	18.49
River flow 1*	5.16	0.91

* *We reduce the input features to the first 6 PCA components.*

References

[1] Ido Kanter. Potts-glass models of neural networks. Phys. Rev. A, 37:2739–2742, Apr 1988.

[2] Joris Mulder and Jean-Paul Fox. Bayesian tests on components of the compound symmetry covariance matrix. Statistics and Computing, 23, 01 2012.

[3] Alejandro Murua and Fernando A. Quintana. Semiparametric bayesian regression via potts model. Journal of Computational and Graphical Statistics, 26(2):265–274, 2017.

[4] WJM Philipsen and LJM Cluitmans. Using a genetic algorithm to tune potts neural networks. In Artificial Neural Nets and Genetic Algorithms, pages 650–657. Springer, 1993.

[5] Shengyang Sun, Changyou Chen, and Lawrence Carin. Learning Structured Weight Uncertainty in Bayesian Neural Networks. In Aarti Singh and Jerry Zhu, editors, Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54 of Proceedings of Machine Learning Research, pages 1283–1292, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[6] Gergely Tibély and János Kertész. On the equivalence of the label propagation method of community detection and a potts model approach. Physica A: Statistical Mechanics and its Applications, 387(19):4982–4984, 2008.

[7] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A java library for multi-label learning. Journal of Machine Learning Research, 12(Jul):2411–2414, 2011.

[8] Alexander Wong, Mohammad Javad Shafiee, Parthipan Siva, and Xiao Wang. A deep-structured fully connected random field model for structured inference. Access, IEEE, 3, 12 2014.

[9] Dong Yu, li Deng, and Shizhen Wang. Learning in the deep-structured conditional random fields. Proc. NeurIPS , vol. 1., pp. 1–8., 01 2009.

[10] Dong Yu, Shizhen Wang, and li Deng. Sequential labeling using deep-structured conditional random fields. Selected Topics in Signal Processing, IEEE Journal of, 4:965–973, 01 2011.

[11] Xiantong Zhen, Mengyang Yu, Xiaofei He, and Shuo Li. Multi-target regression via robust low-rank learning. IEEE transactions on pattern analysis and machine intelligence, 40(2):497–504, 2017.