Double Back-Propagation and Differential Machine Learning

Nonvikan Karl-Augustt Alahassa, Alejandro Murua

▶ **To cite this version:**

Nonvikan Karl-Augustt Alahassa, Alejandro Murua. Double Back-Propagation and Differential Machine Learning. The Ninth Annual Canadian Statistics Student Conference (CSSC), Jun 2021, Ottawa, Canada. hal-03265399

HAL Id: hal-03265399
https://hal.science/hal-03265399

Submitted on 20 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Double Back-Propagation and Differential Machine Learning

Alahassa Nonvikan Karl-Augustt†, Murua Alejandro †

†Universty of Montreal

Introductory Framework

We have introduced a novel (ζ, ε)-Double Back-propagation Scheme (DBS) applicable to any parametric model with convergence properties in terms of Mean Square Error. The DBS indicates that with an optimal number ζ of DBS updates, and appropriate ε learning rate vector for all the model parameters, the Mean Square Error of both training and testing data get to be decreasing, and converge to zero for the training data. The DBS recommends a local Stochastic Gradient Descent (SGD) per observation after the model parameters have been obtained after a first-step estimation with any chosen optimization framework. It has been applied to the Shallow Potts Neural Network Model developed in a previous research by [1], and the results are outstanding. Not the least, we prove mathematically that under an assumption that if there exists a differentiable function that associate covariables (predictors) and target variables (our outputs) as well as their respective local DBS associated parameters, for each observation, we can make the train error and the test error converge to zero simultaneously by applying a dist-NN-h-Taylor Series-PMI model. This last model dictates that we can always differentiate sufficiently the model parameters using a combination of Taylor Approximation Theorem with $h \geq 2$ order with a Perfect Multivariate Interpolation (PMI) framework, and finally, an optimal distance (dist) for a suitable Train-Test covariables association. Our main conclusion is that overfitting, mainly with the convergent DBS optimizer is the beginning of a new type of learning method, as we can still generalize our parametric model with local neighborhood learning with multivariate interpolation and fine tuned empirical differentiation.

The Double backpropagation scheme (DBS)

0.1 The Double backpropagation scheme applied the Shallow Potts Neural Network Model developed in a previous research by [1]

Similarly to our *Iterative projected gradient* (IPG) applied to our model variational parameter λ_w, we found that the Mean Square Error (MSE) of our regression model can also be back-propagated *w.r.t* to each of the model parameter, i.e $\psi = (b^{(1)}, W^{(1)}, b^{(2)}, W^{(2)})$, the main parameters of the network, with $b^{(1)} \in^{l_1}$, $W^{(1)} \in \mathbb{M}_{q\times l_1}$, $b^{(2)} \in^{p}$, and $W^{(2)} \in \mathbb{M}_{l_1\times p}$, $l_0 = q$, $l_2 = p$, $\Sigma \in \mathbb{M}_{p\times p}$ being the variance-covariance matrix of y. First, we know that $y|x, \psi, \Sigma$ is distributed as a multivariate normal distribution with mean $f(y) = f_\psi(y) = \mathbb{E}(y|x, \psi)$, and variance Σ. That is, $p(y|x, \psi, \Sigma) = (2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\{-12(y - f_\psi(x))'\Sigma^{-1}(y - f_\psi(x))\}$. Then, by applying the sampling method based on the Cholesky decomposition, we have :

$$\hat{y}_{est} = f_\psi(x) + L \cdot u = [b^{(2)} + g_1(b^{(1)} + xW^{(1)})W^{(2)}] + L \cdot u \tag{1}$$

such that $L \in M_d(\mathbb{R})$ is a lower triangular matrix such that $\Sigma = LL^T$, and $u \sim N(0, I)$. The *double* backpropagation scheme for the Shallow Gibbs goes as follows:

1. Using the IPG, apply backpropagation method on hyper-parameter λ_w to reduce its Kullback-Leibler (KL) estimation error. Once done, generate an estimate $\hat{\psi}_0$ of $\psi = (b^{(1)}, W^{(1)}, b^{(2)}, W^{(2)})$, using Monte Carlo sampling method from the variational distribution of the parameters.
2. Use equation 1 to backpropagate the $MSE(y_i - \hat{y}_i) = \|y_i - \hat{y}_{est,i}\|^2$ to update $\hat{\psi}_0$ in the Potts cluster and **per observation** as follows:

$$\hat{\psi}_{1,i} \longleftarrow \hat{\psi}_0 - \epsilon_{\psi,0}\frac{\partial MSE(y_i - \hat{y}_{est,i})}{\partial \psi} \tag{2}$$

$$\hat{\psi}_{t,i} \longleftarrow \hat{\psi}_{t-1,i} - \epsilon_{\psi,t-1}\frac{\partial MSE(y_i - \hat{y}_{est,i})}{\partial \psi} \tag{3}$$

The Generalized DBS and its Convergence Property

0.2 A Generalized Double Back-Propagation Scheme (GDBS) for any parametric model

We propose an effective Generalized Double Back-Propagation for any parametric model, augmented with a differential and local neighborhood machine learning framework for almost sure convergence. The General Double Back-propagation Scheme (GDBS) using the Mean Squared Error (MSE), and for any (parametric) model with parameter ψ as :

$$\hat{y}_{est,i} = f_\psi(x_i) \tag{4}$$

Apply to [4] the Double Backpropagation scheme (DBS). To reach convergence, above assignments have to integrate updates for y as follows:

$$\hat{\boldsymbol{y}}_{est,(i,t)} \longleftarrow f_{\psi_{i,t}}(x) + L_{i,t} \cdot u_{i,t} - \epsilon_{\hat{\boldsymbol{y}}_{est,(i,t)}}\frac{\partial MSE(y_i - \hat{y}_{est,i})}{\partial \hat{\boldsymbol{y}}_{est,i}} \tag{5}$$

where $\frac{\partial MSE(y_i - \hat{y}_{est,(i,t)})}{\partial \hat{\boldsymbol{y}}_{est,i}} = 2*(y_i - \hat{\boldsymbol{y}}_{est,(i,t)})$. In practice, we have applied $2*(y_i - \hat{\boldsymbol{y}}_{est,(i,t-1)})$.

So each training data has its own learning rate which is here set as $\epsilon_{\hat{\boldsymbol{y}}_{est,(i,t)}}$. This is valuable for each test data as follows:

$$\hat{\boldsymbol{y}}^{\text{test}}_{est,(i,t)} \longleftarrow f_{\psi_{i,t}}(x^{\text{test}}) + L_{i,t} \cdot u_{i,t} - \epsilon^{\text{test}}_{\hat{\boldsymbol{y}}_{est,(i,t)}}\frac{\partial MSE(y_i - \hat{y}_{est,i})}{\partial \hat{\boldsymbol{y}}_{est,i}} \tag{6}$$

where for the k-th test data x_k^{test} the changes $f_{\psi_{i,t}}$ are taken from the j-th training data y_j which verify:

$$j^{\text{choosen}} =_{x_j \in \text{ Training Set}} \text{Mean}(x_j - x_k^{test}) \tag{7}$$

where the operation Mean(u) for vector u is taken upon all dimension of u. The criteria used in this optimization [7] can be modified for a distance $dist$ for which each test data x_k^{test} is ensured to find an associate x_i^{train} in the training data with :

$$dist(x_k^{\text{test}}, x_i^{\text{train}}) \leq \varepsilon \tag{8}$$

where ε is a very small number. This presented framework will be called $(\boldsymbol{\zeta}, \epsilon_{dbs}) -$ **GDBS**, and augmented with the data Augmentation for Empirical Differentiation (DAED) framework, shall be called the *dist*-NN-(h)-TS-PMI-$(l_1, \boldsymbol{\zeta}, \epsilon_{dbs}) -$ **GDBS**. When the model is truly differentiable [**This is our assumption** $(\boldsymbol{\mathcal{F}}_d)$], the learning with this model is almost surely perfect.

Multivariate Interpolation come into action, as the later can be taken also as a machine *learner*, because it can refine Nearest Neighborhood Train-Test association [7]. To perceive this, let us remind linear interpolation. Linear interpolation usually requires two data points (u_a, v_a) and (v_b, v_b) , and at the point (u, v), the interpolation equation is given by:

$$v = v_a + (v_b - v_a)\frac{u - u_a}{u_b - u_a} \tag{9}$$

To generalize equation [9] to a learning problem, remember the Taylor's theorem for a multivariate function $\boldsymbol{\iota}$ in functions analysis theory [[3], [2]]. We already know the best linear approximation to $\boldsymbol{\iota}$. It involves the derivative $\boldsymbol{D\iota}(\mathbf{a})$ such as:

$$\boldsymbol{\iota}(\mathbf{x}) \approx \boldsymbol{\iota}(\mathbf{a}) + \boldsymbol{D\iota}(\mathbf{a}) \circ (\mathbf{x} - \mathbf{a}) \tag{10}$$

where $D\boldsymbol{\iota}(\mathbf{a})$ is the matrix of partial derivatives of $\boldsymbol{\iota}$ evaluated in the neighborhood of $\mathbf{a}$, and $\circ$ is the dot product between both vectors $\boldsymbol{D\iota}(\mathbf{a})$ and $(\mathbf{x} - \mathbf{a})$. This approximation is linear and represents the first-order Taylor polynomial [[5]]. Multivariate version of Taylor theorem ([4]) is the generalization of approximation [10].

Data Augmentation for Empirical Differentiation (DAED) and Differential Machine Learning

It is not a coincidence that Taylor Approximation theorem is only defined in a certain vicinity or a given neighborhood set! One of the main contribution here is to understand that : Multivariate Interpolation using Taylor theorem is the *refined* generalization of simple Neighborhood Train-Test association. Notice in equation [10], if $\boldsymbol{a} = \boldsymbol{x}^{\text{train}}$ and $\boldsymbol{x} = \boldsymbol{x}^{\text{test}}$, we have $\boldsymbol{\iota}(\boldsymbol{x}^{\text{test}}) \approx \boldsymbol{\iota}(\boldsymbol{x}^{\text{train}})$, when we suppose $\boldsymbol{x}^{\text{train}} \approx \boldsymbol{x}^{\text{test}}$. To avoid that simple approximation, we add more differential terms, as exposed in equation [10].

Taylor Approximation can truly solve any machine learning problem. We illustrate this fact using data augmentation. To take profit from assumption $(\boldsymbol{\mathcal{F}}_d)$ [when it is valid], we need to create more samples from existent training data.

To understand this intuition, remember that the partial derivative of a function $\boldsymbol{\iota}(x_1, \ldots, x_n)$ in the direction x_i at the point $(e_1, \ldots, e_n)$ is defined by:

$$\frac{\partial \boldsymbol{\iota}}{\partial x_i}(e_1, \ldots, e_n) = \lim_{\delta \to 0}\frac{\boldsymbol{\iota}(e_1, \ldots, e_i + \delta, \ldots, e_n) - \boldsymbol{\iota}(e_1, \ldots, e_i, \ldots, e_n)}{\delta} \tag{11}$$

with $\delta \in \mathbb{R}$ has to be a very small real number. Because the closed neighborhood training data j_2 for j_1 required for the differentiation computation is not available in practice in the training data, we can create more data as follows, to compute derivative [11] with almost exact precision:

$$x_{j_1}^{\text{train}} = (x_{j_1}^{\text{train}}(1), ..., x_{j_1}^{\text{train}}(q)) \longrightarrow \begin{cases} (x_{j_1}^{\text{train}}(1) + \delta, ..., x_{j_1}^{\text{train}}(q)) \\ (x_{j_1}^{\text{train}}(1), x_{j_1}^{\text{train}}(2) + \delta, ..., x_{j_1}^{\text{train}}(q)) \\ \cdots\cdots\cdots \\ (x_{j_1}^{\text{train}}(1), x_{j_1}^{\text{train}}(2), ..., x_{j_1}^{\text{train}}(q) + \delta) \end{cases} \tag{12}$$

This method is the reason why **(*dist*)-Nearest Neighbor-(h)-Taylor Series-Perfect Multivariate Interpolation (*dist*-NN-(h)-TS-PMI)** presented previously, is the Perfect fit (or the Perfect learning model) for the Shallow Gibbs Network, summarized in equation [13]:

$$\lim_{l_{1,\text{opt}}, \boldsymbol{\zeta}_{\text{opt}}, \epsilon_{\text{dbs,opt}}, \text{dist}_{\text{opt}}, h_{\text{opt}}} (MSE^{\text{Train}}, MSE^{\text{Test}}) = (0, 0) \tag{13}$$

where MSE^{Train}, MSE^{Test} are the Mean Squared Error of the train and test data respectively, dist_{opt} is the optimal distance for the research of the nearest neighbor in the training dataset for each test data x_i^{test}, h_{opt} is the optimal order of the Taylor approximation for the Perfect Multivariate Interpolation (*dist*-NN-(h)-TS-PMI) model once the $(l_1, \boldsymbol{\zeta}, \epsilon_{dbs}) -$ **DBS** has overfitted the training dataset. $l_{1,\text{opt}}, \boldsymbol{\zeta}_{\text{opt}}$ are respectively the optimal number of hidden neurons, and the optimal number of **DBS** updates. ϵ_{dbs} integrates simultaneously the DBS learning rate vector for all the model parameters, the DBS learning rate for the training data, and the DBS learning rate for the test data. $\boldsymbol{\epsilon}_{\text{dbs,opt}}$ is the optimal one.

References

[1] N. Karl.-A. Alahassa and A. Murua. "Shallow Structured Potts Neural Network Regression (S-SPNNR)". In: *Proceedings of the Edge Intelligence Workshop 2020, Les Cahiers du GERAD G–2020–23* (2020). URL: https://www.gerad.ca/en/papers/G-2020-23-EIW03.

[2] Andrew Browder. *Mathematical analysis: an introduction*. Springer Science & Business Media, 2012.

[3] Mariano Giaquinta and Giuseppe Modica. *Mathematical analysis: An introduction to functions of several variables*. Springer Science & Business Media, 2010.

[4] ZHANG Qian. "High Order Directional Derivative and the Simple Form of Multivariate Taylor Theorem". In: *Journal of Heze University* 2 (2011), p. 4.

[5] Manfred Reimer. *Multivariate polynomial approximation*. Vol. 144. Birkhäuser, 2012.