ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

# Constraint optimization for Echo State Networks applied to satellite image forecasting

Yannic Lieder and Jochen J. Steil

Technische Universität Braunschweig - Institut für Robotik und Prozessinformatik

**Abstract**. The paper proposes to deal with noisy, sparse or short training data sequences by adding domain knowledge to the learning process of Echo State Networks (ESNs). Known constraints like monotony in the output, periodicity or bounds on output values are encoded as inequality constraints on the output weights to be learned. Exploiting that the output of an ESN is linear in the weights, Quadratic Programming is then used to obtain and optimize these. The method is applied to the prediction of pixel values from monthly, noisy satellite images of a short history of five years, thereby enabling the cleaning of images from clouds or snow.

## 1 Introduction

When data is sparse or highly noisy, the embedding of prior domain knowledge into neural networks is of crucial importance to provide the necessary machine learning bias for successful generalization. The traditional way to do this is extensive preprocessing or data augmentation before learning, which lacks both transparency and flexibility and requires hard decisions beforehand. We here follow a different approach to rather use dedicated learning methods, which are being developed particularly for technical domains [1]. Examples are the embedding of e.g. monotony constraints [2] or gain constraints [3] in feedforward neural networks. A wider range of constraints is supported by regression methods in support vector machines [4]. For ESNs, prior knowledge has been used to determine the reservoir topology [5].

In [6], we introduced the Constrained Extreme Learning Machine (CELM) to embed and verify a wide range of constraints in order to insert prior domain knowledge in the learning process of a feedforward network. The CELM exploits that its output is linear in the learning parameter and thus allows for linear inequality constraints on the output. Weights then are found through Quadratic Programming (QP).

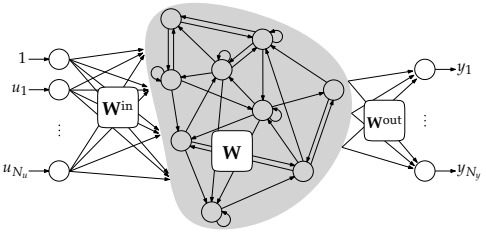



Fig. 1: Echo State Network

The current paper introduces in the same spirit QP for Echo State Networks and encodes constraints on the temporal development of the outputs. In an application to real world satellite data, it shows that thereby forecasting of pixels under highly noisy conditions and very short history is possible and can for instance be used for removing clouds or filling gaps in the image.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

## 2 Constraint Optimization for Echo State Networks

Fig. 1 shows the standard ESN architecture with an input layer, inner reservoir layer and output weights $\mathbf{W^{out}}$, which are the learning parameters, while the inputs weights $\mathbf{W^{in}}$ and reservoir weights $\mathbf{W}$ are randomly drawn from a uniform distribution and fixed [7, 8]. Introducing bias neurons and a leak rate $\alpha$, the update equation in discrete time can be written as

$$\mathbf{x}(t) = (1-\alpha)\underbrace{\mathbf{x}(t-1)}_{\text{previous reservoir state}} + \alpha\underbrace{\sigma\left(\mathbf{W}^{\text{in}}\begin{bmatrix}1\\ \mathbf{u}(t)\end{bmatrix} + \mathbf{W}\mathbf{x}(t-1)\right)}_{\text{reservoir update}}, \quad \mathbf{y}(t) = \mathbf{W}^{\text{out}}\begin{bmatrix}1\\ \mathbf{x}(t)\end{bmatrix},$$

where the linear dependency of the output $\mathbf{y}$ on the output weights $\mathbf{W}^{\text{out}}$ is apparent. Training of the network proceeds by feeding it with $j$ input sequences $u^j(t)$, $t = 1..K_j+T_j$, using the first $K_j$ steps of each sequence as a washout to settle the internal states of the reservoir; by recording of the respective reservoir states $x^j(t)$, $t = K_j+1\,..\,K_j+T_j$; by stacking them for all $j, t$ as rows in a respective regression matrix towards the target outputs $y^{j,\text{target}}(t)$; and finally by solving for $\mathbf{W}^{\text{out}}$ through standard ridge regression [7, 6, 8] minimizing

$$\epsilon_{\text{RMSE}} = \sum_j \frac{1}{N_y}\sum_{i=1}^{N_y}\frac{1}{T_j}\sum_{t=1}^{T}\left(y_i^j(t) - y_i^{j,\text{target}}(t)\right)^2,$$

where $N_y$ ist the dimension of the output vector $\mathbf{y}$.

**Constraint optimization:** Dropping the sequence index $j$ for clarity, consider now a constraint $L(t)$ at some time step $t$ as a weighted linear combination of $i$ previous outputs $y_i(t-h)$, $h = 0...H$ with weights $\gamma_{ih}$ and constant scalar $c$:

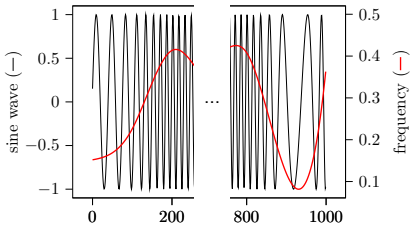
$$L(t) = \sum_{i=1}^{N_y}\sum_{h=0}^{H}\gamma_{ih}\, y_i(t-h) - c. \tag{1}$$

By substituting the linear output equation $y_i(t-h) = \mathbf{W}^{\text{out}}x_i(t-h)$ in $L(t)$ we obtain constraints that are linear in the learning parameters $\mathbf{W}^{\text{out}}$. Such linear combinations include standard difference quotients to approximate derivatives and can express a wide range of constraints, e.g. on the monotony of the output function, bounds on the output values, curvature, periodicity, or to prescribe a dedicated known output value at a particular point in time. We formulate the learning problem under constraints as the Quadratic Program

$$\min_{\mathbf{W}^{\text{out}}} \quad \epsilon_{\text{RMSE}} \quad \textit{such that} \quad L(t) \leq 0, \tag{2}$$

which constitutes the constraint ESN (CESN). Equality constraints, as well as bounds on absolute values can be expressed through two such inequalities. We then use a standard QP solver (Gurobi, https://www.gurobi.com) via Python to obtain the respective output weights $\mathbf{W}^{\text{out}}$.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.




$$|DQ_1(y(t))| \leq 1.1 DQ_1^{\max}$$
$$|DQ_2(y(t))| \leq 1.1 DQ_2^{\max}$$
$$|DQ_3(y(t))| \leq 1.5 DQ_3^{\max}$$

Fig. 2: Left: variable frequency sine wave. Right: constraints (see text.)

**Synthetic example:** We first demonstrate feasibility of the method and provide an illustrative application to show how training with few data can be enhanced through additional knowledge. The task is to predict the time-varying frequency of a sinusoidal input sequence $u(t) = \sin\left(\sum_{i=1}^{t} y(i)\right)$ where data is created by constructing $y(i) = y^{\text{target}}$ from a cubic spline using uniformly sampled points in the range (5/1000, 5/10). Fig. 2 (left) shows a sample of the input $u(t)$ and respective time varying frequency $y(t)$. To make the task harder and pronounce the necessity of reservoir memory, the target is to predict the frequency 50 time steps in the past, i.e. $y^{\text{target}}(t) = y(t-50)$. The constraints (Fig. 2, right) are derived from the known target signal as upper bounds on the first ($DQ_1$), second ($DQ_2$) and third ($DQ_3$) order difference quotients (taken backward in time), and $DQ_i^{\max}$ is the maximum over the input signal.

The training is visualized in Fig. 3. After a first 100 steps of washout, the networks are trained on the next 900 steps while providing the target output $y^{\text{target}}(t)$. Note that small frequencies are hardly present in the training and that 900 time steps for training such memory task is not much. The constraints are applied to the first 4000 time steps excluding the transient washout, whereas they can easily be used also at time steps 1000-4000 where output targets are missing.

We benchmark the constraint optimization approach against two baselines: the same ESN with or without additional post-processing, but without con-

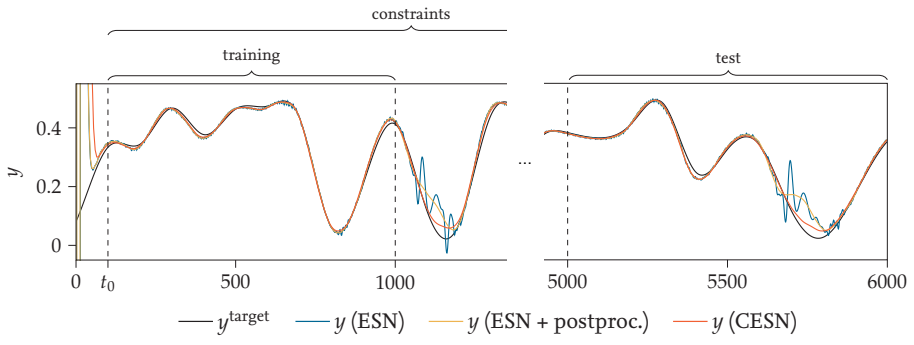



Fig. 3: Washout: 1-100; Training of outputs+constraints 101-1000, constraints only 1000-4000; validation: 4001-5000 (not shown), test: 5001-6000

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

straints. Note that the post-processing is another rather heuristic form of using the domain knowledge to improve performance. It is performed through solving another QP to modify the output $y^{ESN}(t)$ ex-post so that all constraints are satisfied and deviate as little as possible from the original ESN output (in sense of the RMSE). The experiment including hyperparameter optimization was repeated for 30 different network initializations. The best and mean results of each approach together with the found hyperparameters are listed in the table:

| Approach | Abs. Test Error ($\cdot 10^{-3}$) | | Rel. Test Error | |
|---|---|---|---|---|
| | Best | Mean | Best | Mean |
| ESN | 31.7 | $42.1 \pm 6.8$ | 21.9% | 29.1% |
| ESN + post. | 27.0 | $37.4 \pm 4.5$ | 18.7% | 25.9% |
| **CESN** | **11.2** | $\mathbf{16.5 \pm 7.2}$ | **7.7%** | **11.4%** |

| Hyperparameter | Values (Occurrences out of 30 reps) |
|---|---|
| Leaking rate | 0.1, **0.2**(**17**), 0.3(13), 0.7, 0.9 |
| Spectral radius | 0.1, **0.2**(**26**), 0.4(3), 0.6, 0.7(1), 0.99 |
| Input scaling | 0.01, 0.05, 0.1, 0.5(9), **1**(**18**), 2(2), 4(1) |
| Regularization | $\mathbf{10^{-8}}$(**27**), $10^{-5}$(2), $10^{-4}$(1), $10^{-3}$, 1 |
| Reservoir size | 100(6), **500**(**24**) |

This experiment clearly shows that the inclusion of constraints can substantially improve the performance of the network, which here can simply be measured wrt. the known ground truth output. Fig. 3 additionally visualizes that in particular the low frequencies that are hardly present in the input provide a challenge, but can reliably be detected if constraints are added. Note that also postprocessing of a standard ESN with the same constraint improves performance, however performs still worse than our proposed approach. Control experiments further showed that increasing the training data size, e.g. training of full 4000 output values allows to learn the task also for the baseline ESN.

## 3 Application to satellite image data forecasting

In this section, we apply the CESN to processing pixel time series of satellite images which cover specific geographic area of the Harz highland region in northern Germany, specifically its woodland. In our research context, a project towards multi-scale modeling of extreme water and flooding events, it is interesting to assess the "greenishness" of the forests as an indicator of dryness, which in turn determines the wood's capability to retain heavy rains. Images were provided by Planet Labs Inc.[1] in form of monthly aggregated so-called Basemaps, which are preprocessed images of the region and are available for all months starting from January 2016. Ending with December 2020, we obtain a relatively short 60-months image time series. Since we are interested in the appearance of the real dyes of the foliage and plants, we consider both clouds and snow as noise. Another noise factor is foggy and dull weather, which can add a bluish tint to the image as shown in the examples in Fig. 5.

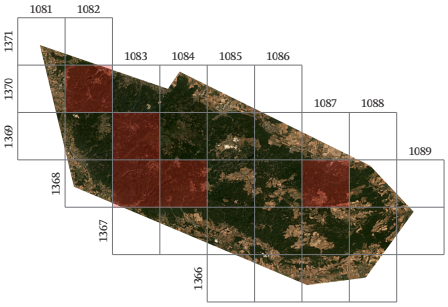



Fig. 4: Basemap in 4096 x 4096 pixel tiles. Used tiles in red.

[1] We thank Planet Labs Inc. for providing the images in this research context free of charge. Basemaps are a commercial product of Planet Labs Inc.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

**Learning Task:** We train and apply a CESN pixel-wise, that is on many independent time series of length 60 of single pixels, which each are defined by the three color channels $y_R$ red, $y_G$ green and blue $y_B$. The task is to predict the pixel of the next month's Basemap.

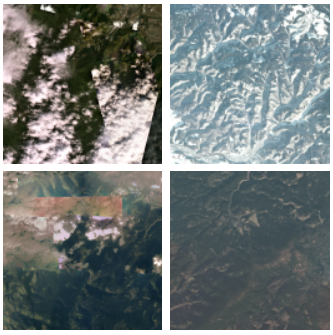


Fig. 5: Noise caused by clouds, snow, fog, processing artifacts.

**Constraints:** We know from the domain that change in "greenishness" is limited from one month to the other and furthermore we assessed that green pixels in our images (as opposed to snow or artifacts) are located in a specific cube in the RGB-space. This knowledge translates to the following constraints: $y_R(t) \leq 0.5, y_G(t) \leq 0.5, y_B(t) \leq 0.35, y_*(t) - y_*(t-1) \leq 0.05, * \in R, G, B$.

**Experiment:** We compare training dataset sizes of 25, 50, 75, 100, 150, 200 pixel sequences, which are uniformly sampled from one of the five tiles shown in Fig. 4. We evaluate the trained model on the other four tiles. The 60-months time series of a pixel is trained, where the first 24 months are used for washout starting from initial zero reservoir state. The months 25 to 59 are trained. While the training dataset size that provides target outputs is variable, we select always in total 1000 sequences including those with targets to embed the constraints. Hyperparameters are determined in advance and the ESN is initialized with regularization: $10^{-8}$, reservoir size 750, spectral radius: 0.99, leaking rate: 1.0 and input scaling: 0.1.

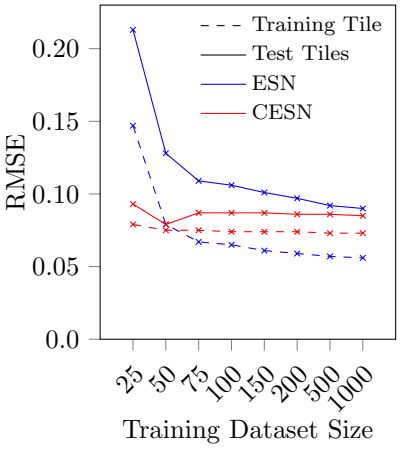



Fig. 6: Error vs trainig set size.

**Results:** Fig. 6 shows quantitative results for the CESN for Dec. 2020 against the baseline of a standard ESN as test performance on sequences of other tiles dependent on the training set size. Whereas Fig. 7 visualizes the result qualitatively in the spatial Basemap image. The standard ESN predicts a snowy image from the earlier years, similar to the rectangular area, and "covers" other parts and tiles with snow, which is visualized through color coding the RMSE deviation from the provided target image. This is reasonable for a plainly data driven method, as it sticks to the provided training data. The CESN, however, is able to "remove" the snow from the rectangular "whitish" image part and rather extrapolate constraint-based some degree of beeing "greenish". In our use case this is desired as we are interested in the dyes of the plants below the snow in training, which is driven by the domain knowledge. This consideration

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

highlights that evaluation of the methods can be non trivial in practice and can not be simply based on error calculations alone. It has to also consider also the domain knowledge and task-dependency.

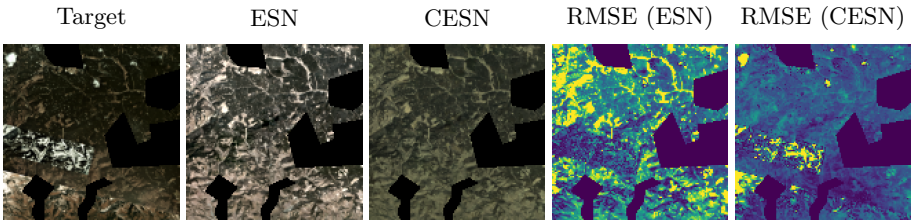



Fig. 7: "Cleaning" a winterly image from snow by adding constraints.

## 4 Conclusion

In this paper, we have shown how to use Quadratic Programming to enrich the training process of an ESN with constraints that reflect specific knowledge about the problem domain. It has been shown that lack of training data or very noisy training can be mediated by such constraints, while a real application on satellite images shows how this can be used for image cleaning. The method uses standard QP solvers, but while these have made enormous progress in recent years, our approach can easily produce very large amounts of pointwise constraints that constitute a significant computational burden, which scales but also varies with the efficiency of the QP solver and the structure of the problem. While in the current work we have presented only constraints in the time-domain of the output. In future work, we would like to extend the approach to include spatial input and spatial constraints amoung input variables as well.

## References

[1] F. Reinhart, K. Neumann, W. Aswolinskiy, J. J. Steil, and B. Hammer. Maschinelles lernen in technischen systemen. In *Steigerung der Intelligenz mechatronischer Systeme*, pages 73–118. Springer, 2018.

[2] Hennie Daniels and Marina Velikova. Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks*, 21(6):906–917, 2010.

[3] Eric Hartman. Training feedforward neural networks with gain constraints. *Neural Computation*, 12(4):811–829, 2000.

[4] Fabien Lauer and Gérard Bloch. Incorporating prior knowledge in support vector regression. *Machine Learning*, 70(1):89–118, 2008.

[5] P. Yu, L. Miao, and G. Jia. Clustered complex echo state networks for traffic forecasting with prior knowledge. In *Int. Conf. Instrumentation and Measurement Technology*, pages 1–5, 2011.

[6] K. Neumann, M. Rolf, and J.J. Steil. Reliable integration of continuous constraints into extreme learning machines. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 21:35–50, 2013.

[7] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[8] Jochen J Steil. Online reservoir adaptation by intrinsic plasticity for backpropagation–decorrelation and echo state learning. *Neural networks*, 20(3):353–364, 2007.