Impact of Regularization on the Model Space for Time Series Classification

Witali Aswolinskiy, René Felix Reinhart and Jochen Steil

Research Institute for Cognition and Robotics - CoR-Lab
Universitätsstraße 25, 33615 Bielefeld, Germany
{waswolinskiy,freinhart,jsteil}@cor-lab.uni-bielefeld.de
http://www.springer.com/lncs

Abstract. Time series classification is an active research field and applicable in many domains, e.g. speech and gesture recognition. A recent approach to classify time series is based on modelling each time series by an Echo State Network and then to classify the time series in the readout weight or model space of these networks. In this paper, we investigate the effect of Echo State Network regularization on the model space. The results show that regularization has a strong impact on the model space structure and the separability of the time series in the model space.

Keywords: model space, echo state networks, reservoir computing, time series classification, time series clustering, regularization

1 Introduction

In time series classification, a label is assigned to a sequence of data points. One main challenge is that the time series can have different lengths. A recent approach is learning in the model space: For each time series a model is trained and the model's parameters are used as features in a consecutive classification stage [2]. Typically, the models are trained to minimize the one-step-ahead prediction error on the time series. The number of model parameters is independent of the time series length, which allows to employ any feature based classifier from machine learning theory in the classification stage.

Echo State Networks (ESNs) [5] are promising candidates for models in this context, because they offer temporal integration of the input, nonlinear computation and quick training. Learning in the model space of ESNs was evaluated for time series classification [2] and clustering [3] with promising results. Here, we investigate the influence of regularization during training of ESNs on classification performance and model space structure.

Regularization is a technique to reduce model complexity in order to prevent overfitting and comes in the context of ESNs often in form of ridge regression. In ridge regression not only the sum of squared residuals is minimized, but also the sum of the squared regression coefficients. This shrinks the coefficients and thereby reduces their variance. Without regularization, and non-orthogonal data vectors, the coefficients can have a large magnitude and be unstable - minor changes in the data can lead to major changes of the coefficients [4, 10]. This is especially important in the context of learning in the model space, where these coefficients form the model space. For a better understanding of the model space we investigate the effects of different forms of regularization on the model space. We show that L2 but not L1 regularization achieves good classification results, and that the modelling error is not necessarily predictive for the classification performance in the model space.

2 Learning in the Model Space of Echo State Networks

A classic ESN architecture consists of a reservoir with recurrently connected neurons and a linear regression readout. The reservoir states $\mathbf{x}$ and the readouts $\mathbf{y}$ are updated according to

$$\boldsymbol{x}(k) = (1-\lambda)\boldsymbol{x}(k-1) + \lambda f(\boldsymbol{W}^{rec}\boldsymbol{x}(k-1) + \boldsymbol{W}^{in}\boldsymbol{u}(k)) \tag{1}$$

$$\boldsymbol{y}(k) = \boldsymbol{W}^{out}\boldsymbol{x}(k), \tag{2}$$

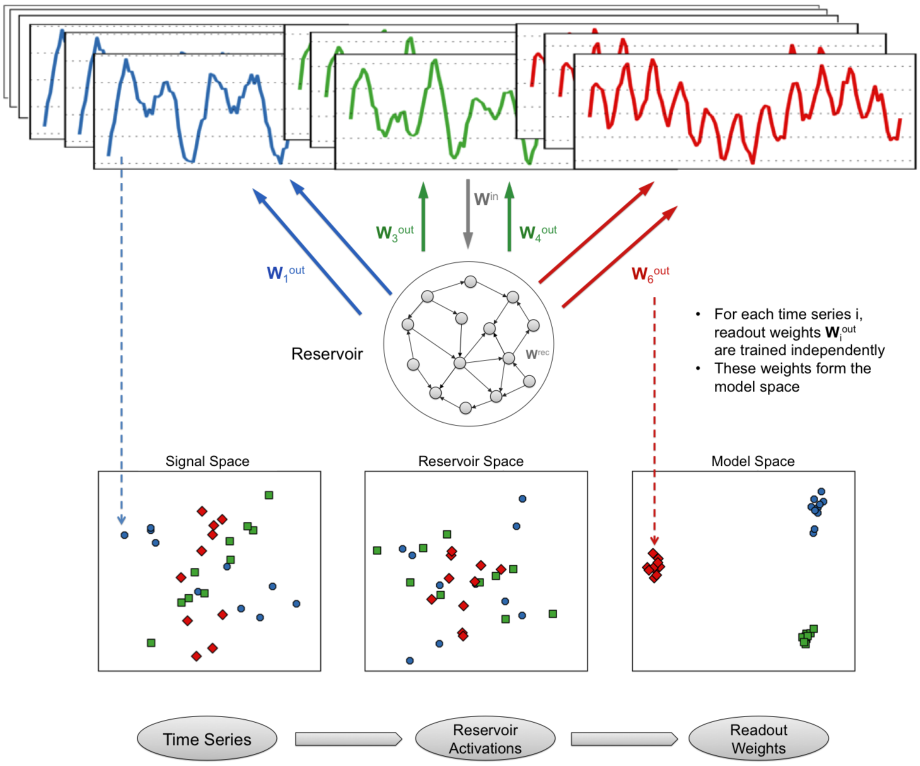
where λ is the leak rate, f is the activation function, e.g. tangens hyperbolicus, $\mathbf{W}^{rec}$ the recurrent weight matrix, $\mathbf{W}^{in}$ the weight matrix from the inputs to the neurons and $\mathbf{W}^{out}$ the weight matrix from the neurons to the readouts $\mathbf{y}$. Only $\mathbf{W}^{out}$ is trained with linear regression - the other weights are initialized randomly and remain fixed.

For classification with ESNs, input time series are fed into the reservoir and the readout is typically trained on the reservoir states with linear regression to predict the class label for each step of the time series. In contrast, in learning in the model space of ESNs, the ESNs are an intermediate step to create a time-independent representation of the time series. Let a dataset consist of N discrete time series $\boldsymbol{u}_i, i = 1...N$ with varying lengths K_i: $\boldsymbol{u}_i(0), ..., \boldsymbol{u}_i(k), ..., \boldsymbol{u}_i(K_i)$. For each time series $\boldsymbol{u}_i$, an ESN is trained to predict from the previous step $\boldsymbol{u}_i(k)$ the next step $\boldsymbol{u}_i(k+1)$ in the time series. The ESNs are trained independently, but share the same reservoir parameters $\mathbf{W}^{in}$ and $\mathbf{W}^{rec}$ in order to create a coherent model space. The prediction error

$$E(\boldsymbol{W}_i^{out}) = \frac{1}{K_i-1}\sum_{k=2}^{K_i}(\boldsymbol{u}_i(k) - \boldsymbol{W}_i^{out}\boldsymbol{x}_i(k-1))^2 + \alpha \left\|\boldsymbol{W}_i^{out}\right\|_L \tag{3}$$

for time series i is minimized, where α is the regularization strength and L is the regularization norm. The resulting readout weights $\boldsymbol{W}_i^{out}$ form the model space, where the classification takes place.

In the model space, arbitrary classification algorithms can be used. This approach will be denoted here as model space learning (MSL) and is visualized in Fig. 1. The example shows the transformation of the data - here noisy sum of sine waves - via regression on the reservoir activations to the readout weights. In this example, the time series and hence the reservoir activations are 100 steps long. The dimensionality of the model space depends on the dimensionality of



- For each time series i, readout weights $\mathbf{W}_i^{out}$ are trained independently
- These weights form the model space

Fig. 1. Learning in the Model Space of ESNs. In the upper part of the diagram, noisy sine waves are shown, which are fed into the reservoir. For each time series a readout is trained to predict the next input. In the lower part of the diagram, the time series, the reservoir echos and the readout weights are visualized with matching colors. The signal, reservoir activation and readout weight matrices were projected to a plane using PCA. In the model space, the time series of different classes are easily separable.

the signal and the number of reservoir neurons. For signals with d dimensions and reservoirs with n neurons the model space dimensionality is $d \cdot n$ or $d \cdot (n+1)$ if a regression intercept is used. For comparison, in the lower part of Fig. 1, each time series is represented as a point in signal space, reservoir space (the space of reservoir neuron activations) and model space. For visualization purposes, the data was projected to two dimensions via principal component analysis. In this example, time series from different classes can be easily separated in the model space.

3 Effect of Regularization on the Model Space

In MSL regularization can occur at two stages. First, while training the ESN models and second, while training a classifier in the model space where the ESN readout weights serve as features. Here, we focus on the effect of regularizing the ESN readout weights $\boldsymbol{W}^{out}$. The regularization strength for ESN training is given by α in Eq.(3).

3.1 L2 and L1 Regularization

In L2-regularization or ridge regression the sum of squared residuals is minimized. In L1-regularization or Lasso regression the magnitude of the weights is minimized [9]. This results in a more sparse weight distribution. We evaluated the effect of regularization in the ESN on the model space on two multivariate datasets from the the UCI repository [7]. Since a direct analysis of model space properties is difficult, we assess the model space structure indirectly by evaluating pattern separability in the classification stage.

The Australian Sign Language Signs (AUSLAN) dataset consists of 2565 samples (27 repetitions x 95 signs), recorded from a native signer using hand position trackers. The sequences are between 45 and 136 steps long and have 22 input dimensions.

The Spoken Arabic Digit (SAD) dataset [6] consists of 8800 samples (10 digits x 10 repetitions x 88 speakers) with 13 input dimensions - Mel Frequency Cepstral Coefficients (MFCCs). The sequences are between 4 to 93 steps long.

Classification error rates were obtained via five times repeated random subsampling, also known as Monte Carlo cross validation [8]. In the AUSLAN dataset, from the 2565 samples, in each fold, 600 randomly selected samples were used for training and the rest for testing. In the SAD dataset the split was half-and-half.

After transforming the time series to model parameters by training the ESNs, in the model space, two classifiers were evaluated: Ridge classifier and support vector machine (SVM) with radial basis function kernel. The ridge classifier is a linear classifier trained with ridge regression: The K class labels are encoded in a 1-of-K scheme and the regressed scores transformed to estimated class labels with the winner-takes-all method. Since the random initialization of the reservoir weights $\mathbf{W}^{in}$ and $\mathbf{W}^{rec}$ affects the performance, only the results of the best out of five reservoirs were used. In order to mitigate the influence of the classifier parameters on the model space analysis, for each α value and classifier, several classifier parameter values were evaluated and from these evaluations the lowest error rate was chosen. The used ESN and classifier parameters are listed in the Appendix.

L2 Regularization: The classification error rates for the datasets at different regularization strengths are depicted in Fig. 2 (A,B). The diagram can be divided in three areas: Overfitting on the left, underfitting on the right and the lowest error rates in the middle. Interestingly, the SVM classifier does not achieve lower error rates than the linear ridge classifier. This suggests, that the considered classification tasks can be solved linearly in the model space.

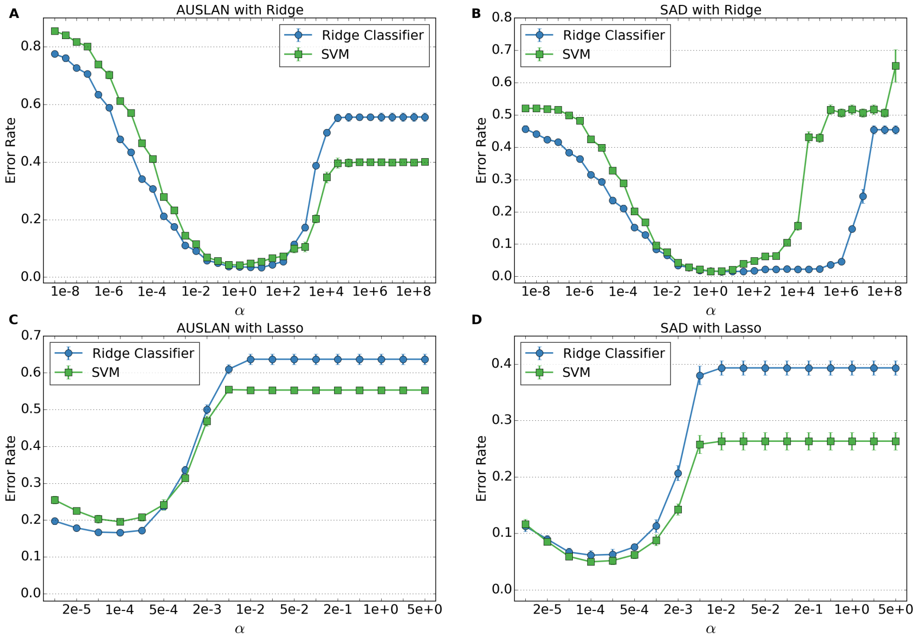


Fig. 2. Classification error rates in the AUSLAN and SAD datasets with ridge (A,B) and Lasso (C,D) regression for different ESN regularization strengths α and two classifiers. The average error rate of 5-fold Monte Carlo cross validation is shown. The very small standard deviations are indicated by error bars.

L1 Regularization: Fig. 2 (C,D) shows the classification error rates for Lasso regression. Compared to Fig. 2 (A,B) the performance is considerably worse. With ridge regression the error rate in AUSLAN was 3.31±0.4% and in SAD 1.5±0.5%. With Lasso regression, the lowest error rate in AUSLAN was 16.6±0.4% and in SAD 5.0±0.6%. These results indicate that sparseness in the model space is detrimental for classification performance. A possible explanation is the nature of L1-regularization: The explanatory variables mostly correlated with the target variable are selected. In MSL, Lasso is executed for each time series independently and thus, for similar but slightly different time series of the same class, different reservoir neurons may be selected. This effect emphasizes the difference between the samples in the model space rather than their similarity. A manual inspection of the model space validated the presence of this effect.

3.2 Difficulty of the Task

In order to investigate, whether there is dependence between regularization and the difficulty of the classification task, we devised a synthetic dataset. The task is to differentiate between time series

$$
\begin{aligned}
u(k) &= \sin(0.2k) + \sin(0.311k) + \sin(0.42k) \\
u(k) &= \sin(0.2k) + (1 - \Delta)\sin(0.311k) + (1 + \Delta)\sin(0.42k)\ .
\end{aligned}
$$

The difficulty of the task is controlled with the parameter Δ: The smaller the value of Δ, the more similar are the respective time series and thus the more difficult the task. The dataset consists of 200 time series of length 100 created by generating 10,000 steps of each sequence and dividing them in 100 parts each. Half of the time series were used for training and half for testing.

Fig. 3 shows how the model prediction error (A) and the classification error in the model space (B) depend on the regularization strength α and task simplicity Δ. The prediction error is the normalized root mean square error (NRMSE) computed for predicting the next input during training of the ESN models and was averaged over all models. With Δ=0, the two time series classes are the same and thus not separable (see first column in Fig. 3B). With increasing Δ the difference between the time series classes increases and the classification error rate decreases. Weak regularization with $\alpha < 10^{-5}$ leads to overfitting: Low model prediction error, but high classification error in the model space (see the upper parts of Fig. 3A and 3B). Regularizing with $\alpha > 10$ leads to underfitting during model training, but the classification error rates remain low until α exceeds 10^7 (see the lower parts of Fig. 3A and 3B). Regularizing with $10 < \alpha < 10^3$ achieves the best results. Noteworthy is that no correlation between the prediction error and classification error rate is observable. This means that for pattern separation in the model space, it is less important how exact a model fits the data.

3.3 Noise and Overfitting

The main goal of regularization is to prevent the learning of noise which leads to smaller errors on the training data, but decreases the generalization capability and therefore causes larger errors on the test data. As the amount of noise in the UCI datasets is unknown, we use again a synthetic dataset to study the effect of regularization on the model space in the presence of noise. The task is to differentiate between time series of the form

$$
\begin{aligned}
u(k) &= \sin(0.2x) + \sin(0.311x) + \sin(0.42x) + \epsilon \\
u(k) &= \sin(0.2x) + 0.7\sin(0.311x) + 1.3\sin(0.42x) + \epsilon\ .
\end{aligned}
$$

This corresponds to the synthetic dataset from the previous section with Δ=0.3 and added noise. During the simulations, the standard deviation σ_ϵ of the Gaussian distributed noise ϵ was varied. Fig. 3C and 3D show the the average of the prediction error and the classification error rate for different regularization strengths α and noise levels σ_ϵ. Best results are achieved with α around 1. Here too, no correlation between the model prediction error and the classification performance in the model space can be observed. Low classification rates are possible even with large model prediction errors, e.g. a NRMSE of 1.0.

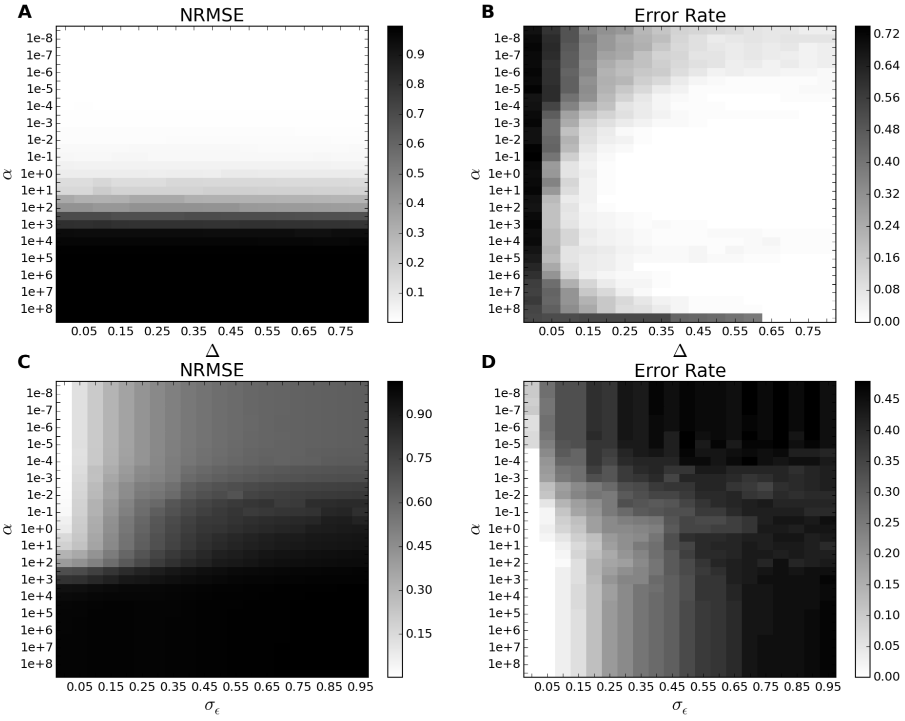


Fig. 3. Average NRMSE for predicting the next value in a time series with the ESNs (A,C) and classification error rate in the model space (B,D). Dark cells denote high values and bright cells low values.

4 Conclusion

In this paper we investigated the effect of regularization during training of ESNs on the classification performance in the readout weight space (i.e. model space) of these ESNs. Regularization improved the performance for more difficult tasks as well as in the presence of noise. L2 regularization performed considerably better then L1 regularization. In conclusion, L2 regularization is more suitable for MSL, and the model prediction error can not be used to estimate the expected classification performance in the model space.

5 Appendix

Preprocessing: All datasets were scaled to the range $[-1/d, 1/d]$.

Reservoir Parameters: n: Number of reservoir neurons, Input Scaling: Scaling of the weights from the input to the reservoir; Rec. Connectivity: Density of the recurrent weight matrix $\mathbf{W}^{rec}$, Rec. Scaling: Spectral radius of the recurrent weights $\boldsymbol{W}^{rec}$, λ: Leak rate of the neurons, Bias Scaling: Scaling of the neuron biases.

- Synthetic: n: 50, Input Scaling: 1, Rec. Connectivity: 10%, Rec. Scaling: 0.9, λ: 1, Bias Scaling: 0.1
- AUSLAN: n: 100, Input Scaling: 1, Rec. Connectivity: 10%, Rec. Scaling: 0.9, λ: 1, Bias Scaling: 0.1
- SAD: n: 50, Input Scaling: 2, Rec. Connectivity: 10%, Rec. Scaling: 0.9, λ: 0.7, Bias Scaling: 0.1

Classifier parameters: For both synthetic datasets, a linear regression classifier without regularization was used. In AUSLAN and SAD:

- Ridge Classifier: Ridge from 1e-5 to 1e+5 in logarithmic scale.
- SVM Classifier [1]: Penalty C from 0.1 to 10000 in logarithmic scale, Kernel coefficient gamma from 0.01 to 100 in logarithmic scale.

Acknowledgments. This project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition "it's OWL" (intelligent technical systems OstWestfalenLippe) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
2. Chen, H., Tang, F., Tino, P., Yao, X.: Model-based kernel for efficient time series analysis. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 392–400 (2013)
3. Chen, H., Tino, P., Rodan, A., Yao, X.: Learning in the model space for cognitive fault diagnosis. IEEE Transactions on Neural Networks and Learning Systems 25(1), 124–136 (2014)
4. Hoerl, A.E., Kennard, R.W.: Ridge regression: applications to nonorthogonal problems. Technometrics 12(1), 69–82 (1970)
5. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. GMD Technical Report 148, 34 (2001)
6. Kadous, M.W.: Temporal classification: Extending the classification paradigm to multivariate time series. Ph.D. thesis, The University of New South Wales (2002)
7. Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
8. Picard, R.R., Cook, R.D.: Cross-validation of regression models. Journal of the American Statistical Association 79(387), 575–583 (1984)
9. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)
10. Vinod, H.D.: A survey of ridge regression and related techniques for improvements over ordinary least squares. Review of Economics and Statistics pp. 121–131 (1978)