PML - Report

Plastic State Machines: From Neuroscience to Machine Learning

Thomas Rost[1]

[1]Neuroinformatics, Institut of Biology Freie Universitt Berlin, Germany

April 30, 2014

Contents

1 Introduction

The activity measured in the mamalian neo-cortex is highly variable. This variability is found on all time-scales, from the seeminlgy stochastic emission of individual action potentials, to slow fluctuations in activity level in the order of minutes [1]. It remains unclear, how cortical information processing takes place in the presence of those fluctuations. While it has formerly been assumed to be noise that is somehow averaged out, more recent findings suggest that it may be an integral part of the cortical algorithm. Experimentalists have found that the varaibility across experimental trials is reduced on presentation of a stimulus [2]. Analysis of these dynamics suggest that the variability can be split into two components. On a short time-scale, the bombardment with incoming action potentials from thousands of other neurons causes the timing of individual spikes to be highly variable. On a longer time-scale, the firing rate of the neurons seem to change between experimental trials [3]. The meaning of these slow changes is not clear.

A recent study with newborn ferrets has shown that the spontaneous activity in the visual cortices changes with development [4]. Before the animals first open their eyes the activity is random but becomes more and more structured as visual experiences form. In the developed visual system, the spontaneous activity shows slow transitions between patterns very similar to those evoked by visual stimuli. The authors interpret these changes in activity as a possible encoding of priors for certain stimuli. This presents an interesting explanation for the slowly varying variability component.

The variability in cortex is usually simulated with large networks of spiking model neurons for wich the incoming excitatory and inhibitory connections balance out so that very few spikes are emitted on average. These network can however not capture the modulation in the firing variance observed experimentally. This has however been recently achieved by introducing a clustered connectivity structure into the network [5]. The slowly varying variability component now arises from transitions between activations of different clusters. Upon stimulation, some clusters are excited, while others are inhibited, thereby interrupting these transitions and reducing the variability.

The aim of the current work is to examine whether the above effects can be utilised in a machine learning context, i.e. whether the attractor dynmaics induced by networks with clustered connectivity can aid information processing. The starting-point for this study is the concept of reservoir computing. The two most prominent representatives of this technique are Liquid State Machines (LSM) [6] and Echo State Networks (ESN) [7]. While LSMs are implemented with spiking integrate-and-fire models and ESNs use preceptron type units, both methods employ the same principle. A time varying input signal is inserted into a large randomly connected reccurent neural network. The dynamics in this reservoir performe temporal and spatial expansions of the data which are useful for classification or regression. Classification is then performed with memory-less read out units which *see* only one time step of the reservoir ac-

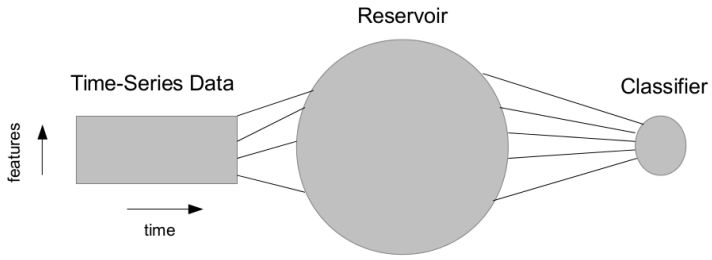

Figure 1: Classification setup. The time series data is fed into the computing reservoir. Classification is performed on individual time-steps of the reservoir activity.

tivity.

The first part of this study will deal with a comparisson of the classification performance of several different reservoir setups. In the second part, plasticity will be introduced into the equation to investigate whether the observations from neuroscience can be transfered to reservoir computing.

2 Methods

Throughout this work, reservoir computing is used in a classification setting (see figure 1). Time series patterns are fed into the computing reservoir. Classification is performed using logistic regression on single timesteps of the reservoir output. Only binary classification was considered here and the performance was evaluated by using the ROC-AUC score, including all time samples. Different strategies for the construction of the reservoir were examined.

2.1 Neuron Models

Two different neuron models were employed. The first one is a simple perceptron implementation as used in the ESN. It has no memory and its state x is updated as the weighted sum of the activities of the other reservoir units and the input [7]:

$$x(t+1) = \tanh(Wx(t) + wu(t+1)) \tag{1}$$

where W is the recurrent reservoir connectivity matrix, w represents the input weights and u is the input signal.

To avoid the additional complications of spiking neuron simulations, a rate model for modelling balanced networks as introduced by Abbot and Sompolinsky [8] was used to mimik the LSM. Here, the unit's state x is characterised by

a differential equation of the form:

$$\tau \frac{dx_i}{dt} = -x_i + g \sum_{j=1}^{N} W_{ij} r_j + I_i \tag{2}$$

g is a scale factor controlling the overall variance of the weights W and I_i is the input to unit i. The units do not influence eachother directly through their state x but through the rate $r = R_0 + \Phi(x)$. Here, R_0 is the baseline rate and the transferfunction $\Phi(x)$ is:

$$\Phi(x) = \begin{cases} R_0 \tanh(x/R_0), & \text{for } x \leq \\ (R_{max} - R_0) \tanh(x/(R_{max} - R_0)), & \text{for } x > 0 \end{cases} \tag{3}$$

Hence r can vary between 0 and R_{max} and returns to R_0 with time constant τ when no input is present. The non-zero baseline-rate R_0 prevents the activity in the balanced network from dying out.

2.2 Connectivity Types

The aim was here to examine the feasibility of clustered connectivity matrices for reservoir computing. For comparisson, two different connectivity types were also included.

The ESN reservoir is constructed with a sparse connectivity matrix with its non-zero elements drawn from a uniform distribution with zero mean. It is then rescaled so that its largest absolute Eigenvalue (the spectral radius) lies around unity. The exact value of the spectral radius is then tuned for performance on specific tasks [7].

In the publication from which ther rate model originates, the authors examine the variability dynamics of spontaneous versus stimulus-evoked activity [8]. For completeness, the connectivity type they employed was also included here. Here, a dense matrix was drawn from a normal distribution and rescaled to have variance $1/N$.

In both the above cases, no distinction was made between inhibitory and excitatory units. The clusterd connectivity type described by Lithwin-Kumar et al. [5] is more closely inspired by neuroscience. The network is divided into inhibitory and excitatory units with a ratio of one to four. The overall connection probability between excitatory units is 0.2, while those excitatory-inhibitory, inhibitory-excitatory and inhibitory-inhibitory connection ocur with a probability of 0.5. The excitatory units are grouped into a number of clusters. The ratio of connection-probabilities within clusters to across clusterd is controlled by a parameter (R_{EE}). At $R_{EE} = 1$, the probabilities are equal and with increasing R_{EE} the clustering gets stronger. This connectivity will in the following be reffered to as the LK-connectivity. Sample weight-matrices are shown in figure 2.

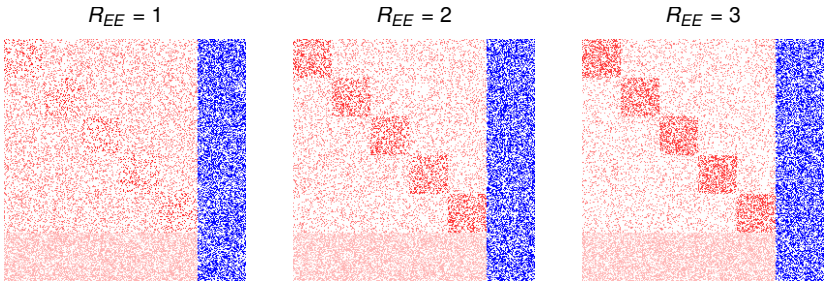

Figure 2: LK-connectivity matrices for increasing clustering strength R_{EE}. Inhibitory weights are displayed in blue, red means excitatory.

2.3 Data

In the first part of the study, the classification performance of the different reservoir types was compared on a multivariate time-series dataset. For this purpose, artificial data was generated guided by the process described in [6]. Since the LSM is spiking, Maass et al. drew random spike patterns as class templates. These were then jittered and embeded into longer sections of spiketrains with the same statistics to for the individual samples for each class. This approach was here adjusted to the use of rate-models by smoothing these spike patterns with narrow kernels and adding some rectified gaussian noise to them. For the training samples, the templates were placed in the background spiketrains at varying positions, to create a more difficult/realistic classification scenario. For the test samples, the templates were centeres, to enable averaging over the classifier outputs. The data is illustrated in figure 3.

In the second part was to simplify the problem and eliminate the temporal component from the data. For this purpose the MNist hand-written digit dataset was used. Two classes were chosen (digits 8 and 9) and the patterns were repeated to form static *time-series* of 50 samples lenght. Again, gaussian noise was added to the data.

2.4 Learning Rules

In the second part, plasticity is introduced into the input-weights of the reservoir. For this purpose, an unsupervised Hebbian learning rule has been used. For rate models with positive unit activities, the covariance rule seems suitable. Here, the weight updates are computed from the product of the deviations of the pre and post synaptic units from their average [9]:

$$\Delta w = \gamma(x(t) - \langle x \rangle)(y(t) - \langle y \rangle) \tag{4}$$

x and y are the pre and post synaptic activities. γ is a small learning rate and $\langle\rangle$ denotes the average over all past values of the activations.

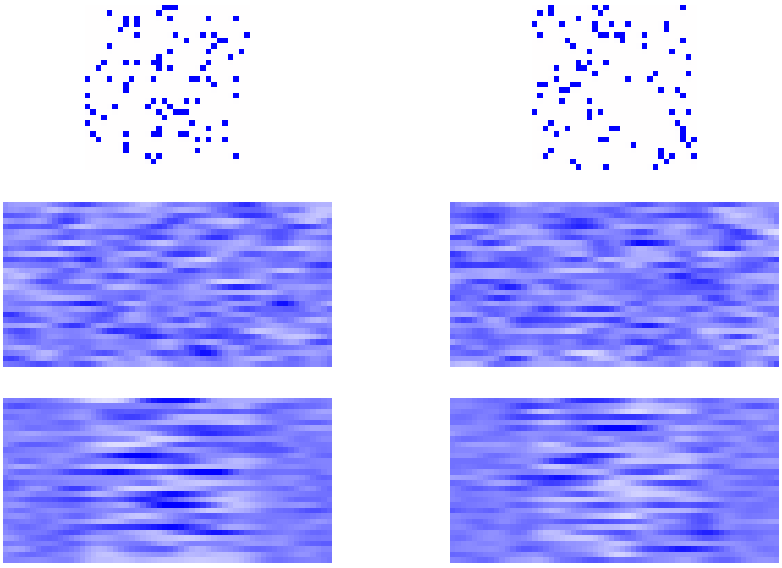

Figure 3: Generation of artificial data. The top row shows the template for the two classes. In the middle, sample train patterns for each class are displayed. On the bottom, the averages over all test samples for both classes are shown.

Hebbian learning rules have the property, that weight growth is exponential over time [9]. To prevent this, a normalisation term is introduced. Oja's rule is multiplicative and has the advantage that it is a local learning rule, i.e. it uses only information from two units:

$$\Delta w = \gamma(x(t) - \langle x \rangle)(y(t) - \langle y \rangle) - w(y(t) - \langle y \rangle)^2 \tag{5}$$

3 Results

3.1 Exploring Reservoir Types

To explore the performance of the different reservoir types, various combinations of neuron model and connectivity matrices were tested on the artificial dataset. The following tables give an overview of the combinations tested and their parameters.

In total four different combinations were tried. Looking at tables 1 through 4, it becomes apparent, that some of the models have many tuneable paramters. While for the pure ESN some guidelines exist on choosing the parameters [7],

ESN	the standard Echo State Network
`N`	the number of units
`sr`	spectral radius of the weight matrix
`pc`	connection probability of the weight matrix
`input_sparsity`	connection probability of the input weights
`input_scale`	scale factor for the input weights

Table 1: Parameters for the ESN-reservoir.

.

ESNLK	ESN-type network with Lithwin-Kumar connectivity
`N_E`	the number of excitatory units
`N_I`	the number of inhibitory units
`sr`	spectral radius of the weight matrix
`n_cluster`	the number of excitatory clusters
`R_EE`	the claustering strenght parameter
`input_sparsity`	connection probability of the input weights
`input_scale`	scale factor for the input weights
`P_EE`	excitatory-excitatory connection probability
`P_IE`	inhibitory-excitatory connection probability
`P_EI`	excitatory-inhibitory connection probability
`P_II`	inhibitory-inhibitory connection probability
`w_in_out_ratio`	the ratio of connection strength between in-cluster and across-cluster weights

Table 2: Parameters for the ESNLK-reservoir.

.

ASmodel	Abbott-Sompolinsky neuron with dense gaussian connectivity
`N`	the number of units
`tau`	the time constant of the neuron model
`scale`	the scale factor of the weight matrix (g)
`input_scale`	scale factor for the input weights
`R0`	the baseline rate
`Rmax`	the maximum firing rate

Table 3: Parameters for the ASmodel-reservoir.

.

ASLKmodel	ASmodel with Lithwin-Kumar connectivity
N_E	the number of excitatory units
N_I	the number of inhibitory units
tau_e	the time constant of the excitatory units
tau_i	the time constant of the inhibitory units
scale	the scale factor of the weight matrix (g)
input_scale	scale factor for the input weights
R0	the baseline rate
Rmax	the maximum firing rate
n_cluster	the number of excitatory clusters
R_EE	the claustering strenght parameter
input_sparsity	connection probability of the input weights
input_scale	scale factor for the input weights
P_EE	excitatory-excitatory connection probability
P_IE	inhibitory-excitatory connection probability
P_EI	excitatory-inhibitory connection probability
P_II	inhibitory-inhibitory connection probability
w_in_out_ratio	the ratio of connection strength between in-cluster and across-cluster weights

Table 4: Parameters for the ASLKmodel-reservoir.

.

for the ASmodel and the LK-connetivity type it is unclear which parameters may be suitable for classification.

It was therefore decided to conduct a large parameter search. For each model type, parameter ranges were selected which seemed feasible. Since this space by far exceeded the size covarable by a complete grid search, parameter combinations were then drawn randomly. Because of the randomness involved in generating the reservoir networks, ten repeated evaluations were performed for each parameter set. The networks were trained on the test set and their auc-score was calculated on the validation set. To give the search a bit more direction and to ensure that the lists of possible values given initially for each parameter would not limit the results a second iteration was performed where the parameters of models with supperior performance were sligtly modified and reevaluated. This effectively ammounts to a form of random gradient ascent on the parameters.

To get an impression of the influence of individual parameters on the classification performance, the scores were plotted against each parameter separately. These plots are shown in figures 4 through 7. The dashed line in each plot represents the *Dummy*-score, i.e. the score achieved by pure logistic regression without the use of a reservoir. Each plot contains all values of the other parameters. The representation is therefore not usefull for showing effects of certain parameter combinations but simply shows whether individual

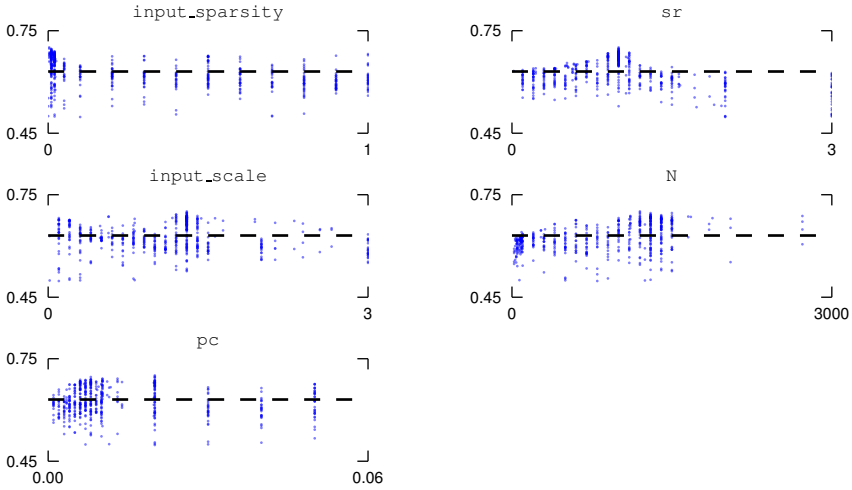

Figure 4: Illustration of the parameter search for the ESN reservoir. Each box contains a plot of the classification score from 10 repetions on the validation set over the value of a particular parameter. The dashed black line represents the average score of logistic regression without the use of a reservoir.

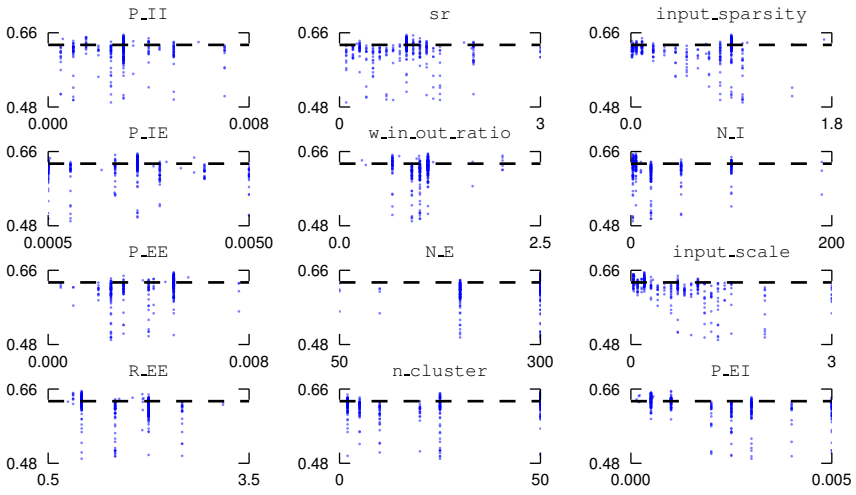

Figure 5: Parameter search for the ESNLK reservoir

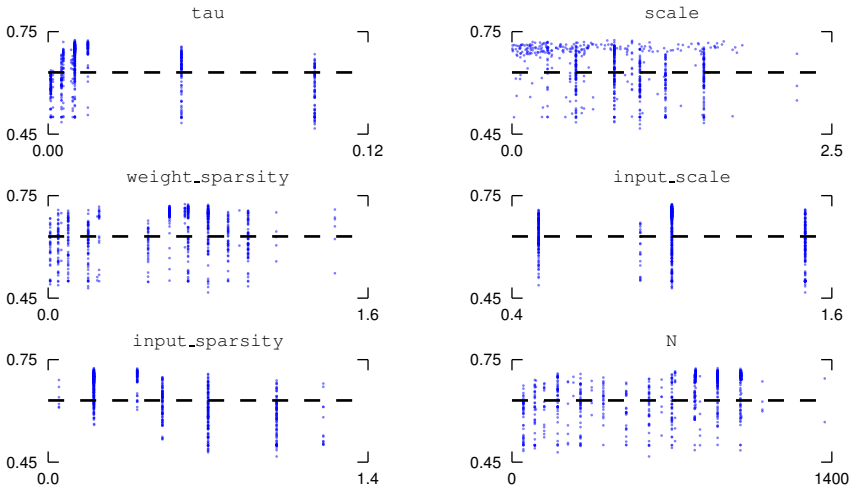

Figure 6: Parameter search for the ASmodel

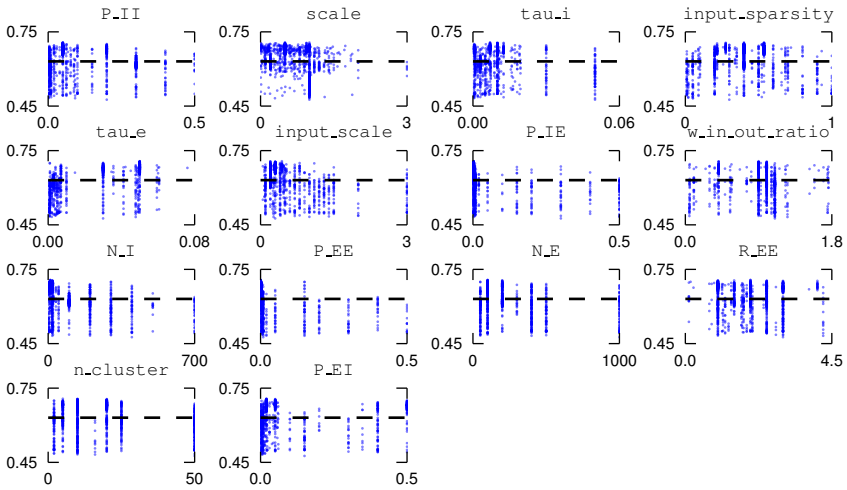

Figure 7: Parameter search for the ASLKmodel

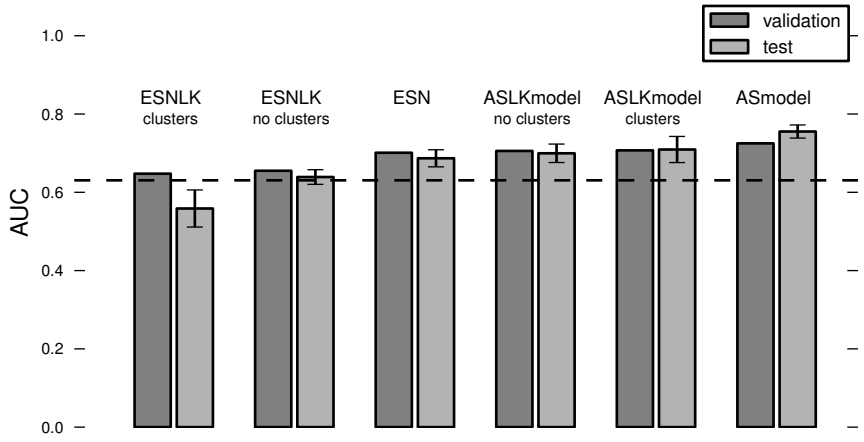

Figure 8: The best scores resulting from the parameter search for each model. The error bars for the test-set represent the standard deviation of ten repetitions.

parameters have a dominant effect on the performance.

It can be read from those figures that the ESN is the most *well behaved* model. It achieves relatively good performance for wide parameter ranges and its performance seems to be best around the parameter settings suggested in [7] (sr=1) ,pc=1%,input_scale=1). for the other models the parameters seem to be much more interdependent and no clear trends can be observed in any of the parameter plots.

To investigate the influence of clustered connectivity on the reservoir performance, it was therefore decided to select the parameter set scoring best on the validation set for each model type and compare their scores on the test set. When LK-connectivity was used by a model, the scores for R_EE=1 and R_EE$>$1 were treated separately. The results of this are shown in figure 8. The models are ordered for increasing validation scores from left to right. All models could beat the pure logistic regression which is shown as the dashed line in the figure. For all models except the ESNLK, the test scores are very similar to the validation performance. Where LK matrices were used, the performance is marginally better if clustering was turned on. The most iteresting result to be read from this image is however, that the AS-neuron model outperfomed the perceptron of the ESN. This indicates that the internal *memory* of the differential equation model is helpful for time-series classification.

To illustrate the effect of the reservoirs on the classification output, time resolved auc-scores were calculated on the test set for each of the models shown in figure 8. In figure 9 it can be seen that the effect of the reservoirs is to pro-

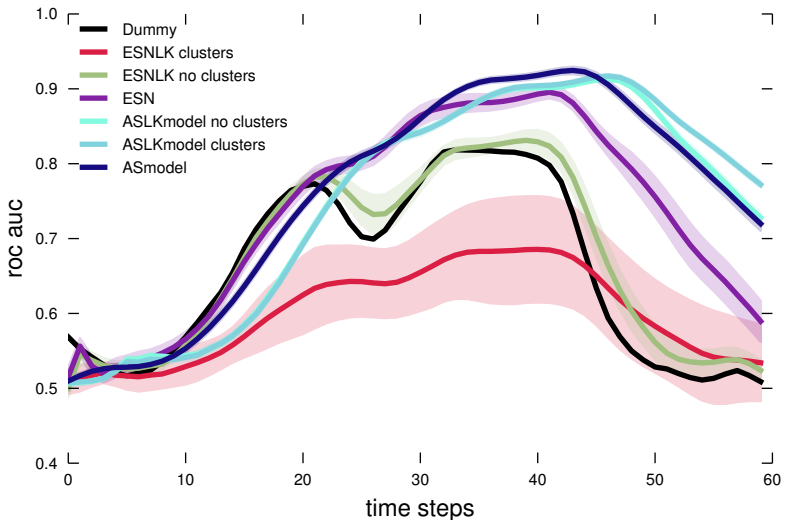

Figure 9: Time courses of the classification scores on the test set for the best performing models of each type. Shaded areas indicate the standard deviation of the repetitions.

long the presence of information even after the input pattern has disappeared. While the rise of the auc-score starts at the same time as for the pure logistic regression (shown in black in the figure), it rises higher and stays up longer for the reservoirs using the AS-model. The time constants in the model do therefore not slow down the *recognition* process at the onset of the pattern but at the same time prolong its presence in the reservoir.

3.2 Introducing Plasticity

The previous section has mainly shown two things. The AS-model is most suitable for time-series classification and clustered connectivity does not really increase the performance. Looking back at the initial motivation from neuroscience, this is not very suprising. The working hypothesis is that the activity of certain clusters is related to specific stimuli or input patterns. In the setup presented in the previous section this is unlikely to happen because the reservoirs and input weights are generated randomly and different input patterns have no reason to prefferentially activate certain clusters or combinations of clusters. It can be argued that in clustered structure is present in the brain on many spatial scales, e.g. in the form of cortical columns. It is also known that the connections between cortical neurons are plastic. In this section it will therefore be attempted to achieve pattern specific activation of clusters by introducing a hebbian learning rule to the input weights of the reservoir.

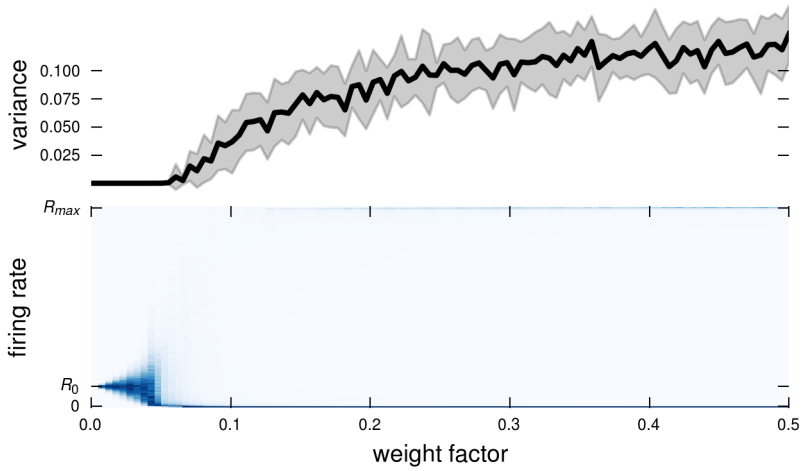

Figure 10: Influence of weight scale on the network dynamics. The upper panel shows the accross trial variance for 100 trials averaged over all units. The shaded area represents the standard deviation over 20 repeated realisations of the the network. The lower plot shows histograms of of the pooled unit firing rates for each scale value.

It was shown that the performance of the reservoirs depends heavily and relatively unpredictably on the specific combination of the network parameters. In this section the subject of optimal classification is therefore neglected and a more qualitative approach is taken. Also, the temporal aspect of the input patterns is dropped to simplify the task of testing whether specific paterns can activate specific clusters. The noisy MNist numbers prepared as described above were taken as the classification task. An ASLKmodel is used with parameters as in the original publications of the components ([5] and [8]) with the exception of the network size. The network is scaled down from 5000 to 500 (400 excitatory, 100 inhibitory) units for reasons of computational cost. To achieve th same number of units per cluster, the number of clusters has been set to 8.

The starting point for this endeavour is that the network has to have suitable dynamics when no input is present. This means that the firing of individual units must vary chaotically while the activity of entire clusters should change on slower time scales. In the first step, the scale parameter of the reseroir weights is tuned. Its influence on the spontaneous dynamics is shown in figure 10 and 11.

The starting point for this endeavour is that the network has to have suitable dynamics when no input is present. This means that the firing of individual units must vary chaotically while the activity of entire clusters should change on slower time scales. In the first step, the scale parameter of the reseroir

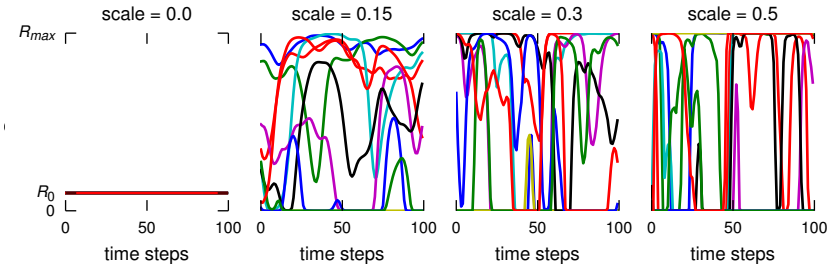

Figure 11: Individual unit activities for different values of weight scale.

weights is tuned. Its influence on the spontaneous dynamics is shown in figure 10 and 11. The top panel of figure 10 shows that the variance of individual units accros trials (i.e. repeated simulations with the same network) is zero up to a scale factor of about 0.05 and then increases to reach a plateau around 0.4. The histogram plot in the bottom panel reveals that at small values of the scale factor, the influence of the recurrent weights is too small to produce chaotic behaviour and all units idle about with smal firing rates. At higer scale factors the units are driven to have predominantly very high or low rates with short transitions between them. This is illustrated for individual units in figure 11. At scale=0, all units are inactive at R_0. At scale=0.5, all units are eiter at 1 or 0 with rapid transitions between the extremes. For the pesent purpose it seems suitable to pick a state where the units are not yet driven into saturation, to leave room for additional input from the input patterns. For this reason, a scale factor of 0.15 is assumed from now on.

Then next step is to tune the clustering strenght so that the average cluster activities vary slowly and separately over time. Figure 12 shows the average firing rate for all units in each of the 8 clusters for different values of `R_EE`. Note the differnce in time scale compared to figure 11. At `R_EE`=0, no difference is observable between the clusters. All units follow some sort of oscillation, which is commonly observed in balanced networks. As the cluster strength increases, the cluster activities start to diverge from eachother and at `R_EE`=3, the cluster activities are spread about the whole firing rate range and switch between high and low activity states. This is the expected behaviour and `R_EE`=3 is assumed from now on.

Now that the spontaneous network dynamics are tuned, a suitable magnitude for the input has to be found. For this purpose the scale factor of the input weights was varied. Figure 13 shows the result of this tuning process. The bottom panel shows the classification performance for each setting of input strenght. Obviously the classification is at chance level for in input strenght of 0. It then peaks at 0.5 and decreases again for larger values. The plots in the uper half of the figure show sample unit dynamics for the respective input sterngths. When the input becomes too strong, the units tend to be driven into saturation, thereby decreasing the information capacity. In the following, a value of 0.5 is used.

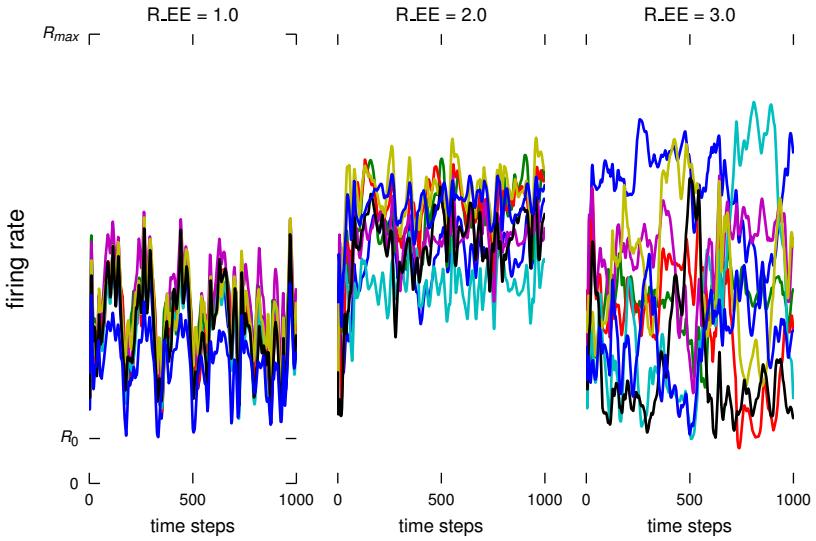

Figure 12: Cluster activities for increasing cluster strength. Each line represents the average firing rates for all units in the same cluster.

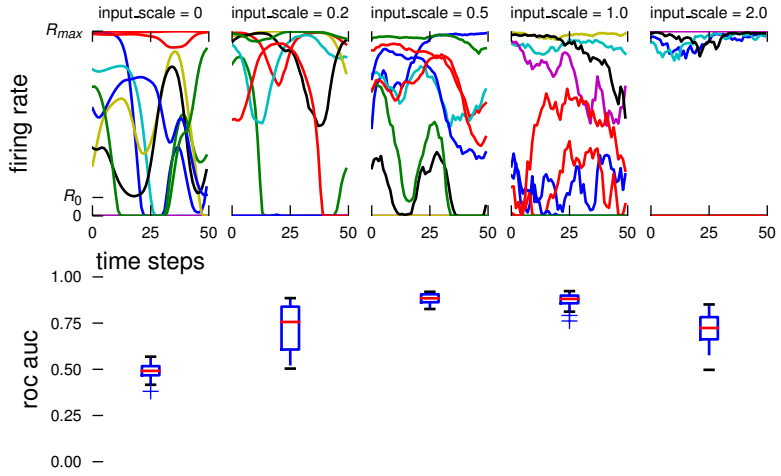

Figure 13: Influence of input strenght. the top panels show sample unit rates, the bottom plot shows distributions of validation scores over 10 repeated realisations of the networks.

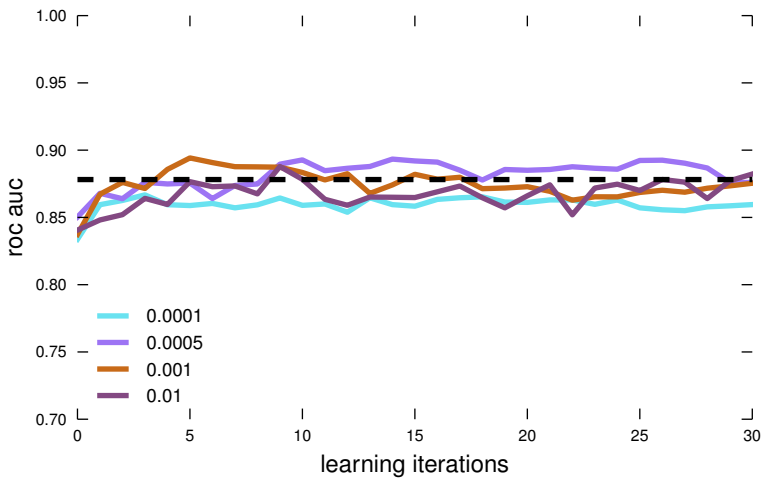

Figure 14: Model performance on the validation set vs training iterations for different learning rates. The dashed black line indicates the score of pure logistic regression.

Having tuned the network to an operating point compatible with the initial hypothesis, the learning rule was now introduced to the input weights. To prevent uncontrolled growth of th weights, the normalisation term according to Oja was used in the covariance learning rule. The only parameter of this learning rule is the learning rate γ. To investigate the effect of learing, the auc-score on the test set was evaluated after training a classifier on the output of the reservoir on the train set before learning. A learning iteration consisted of passing the train set through the network, updating the input weights after each time step. After each iteration, the performance on the test set was again evaluated. The results for several values of γ averaged over 10 repetitions of the process are shown in figure 14. The figure shows that if the learning rate is not too small, the plasticity rule does slightly improve the performance of the performance of the reservoirs.

To investigate how learning affects the dynamics of the networks, a closer look was taken using a learning rate of 0.0005. According to the hypothesis, learning should lead to selective activations of specific clusters. One indication for this would be an increase in correlation between the classification target and the average activity of certain clusters. That this seems to be the case can be seen in figure 15. Here, the average absolute correlation of the individual cluster activities with the classification target indeed seems to increase. The values are quite small. However, closer inspection as shown, that single cluster become highly correlated to one class, while most others have correlations close to zero. According to the hypothesis, the selective activation of clusters

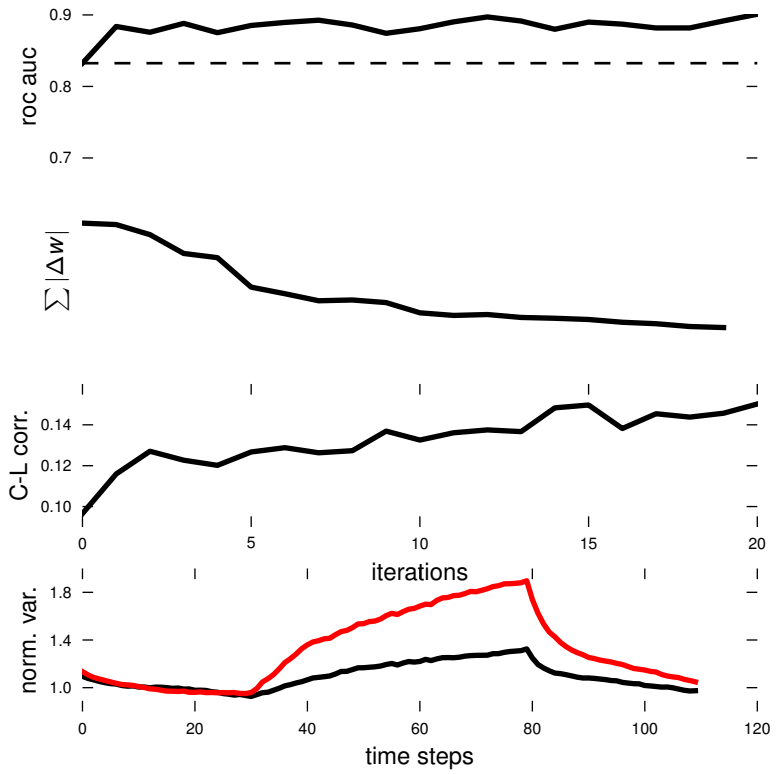

Figure 15: Effect of learning on network dynamics averaged over 10 repetitions. The top pannel shows the classification performance over the training iterations. The second plot shows the total absolute change in input weights and in the third pannel shows the average absolute correlation between cluster activities and the classification target. In the bottom plot, the normalised trial-to-trial variances of all excitatory units are plotted over the course of the noise-padded input. The pattern occurs between timesteps 30 and 90. Black = before learning, Red = after learning

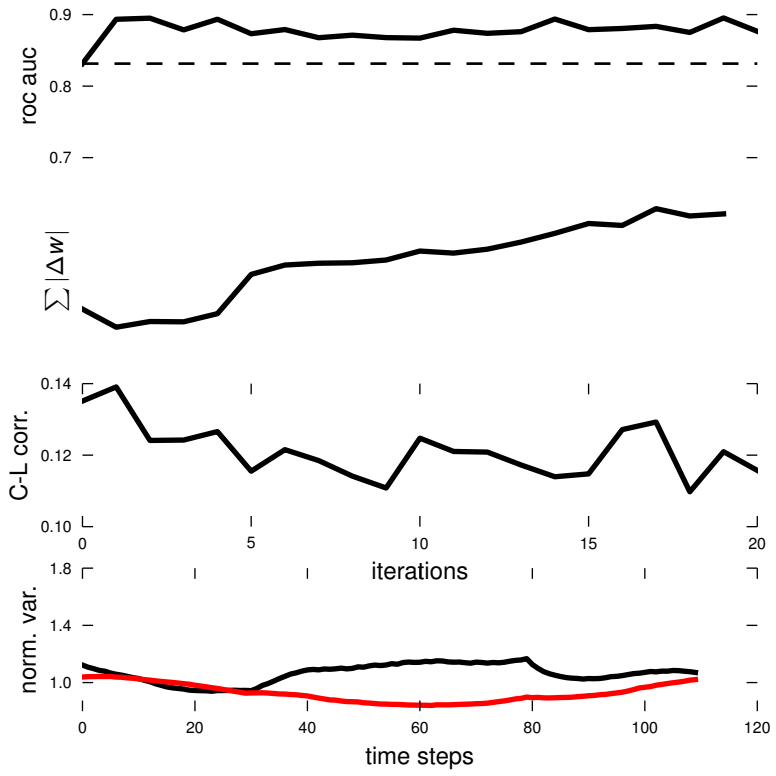

Figure 16: Same as figure 15 but with a learning rule with no normalisation term.

should also reduce the trial-to-trial variance of the units when a pattern is presented. To test this, the test data was padded with noise of the same statistics as the on added to the input patterns. The padded data was processed by the networks and the trial to trial variances were computed. Because the amplitudes of those variances vary between network realisations, the variances were divided by the mean value of the pre-pattern interval (i.e. the response to the noise adding only). The results before and after learning are shown in the bottom pannel of figure 15. Unfortunately, learning even seems to increase this variance.

In a last experiment, the normalisation term was turned off in the learning rule and the same analyses as before were performed. The results of this are shown in figure 16. As expected, the mean weight change now grows with the number of training iterations. Interestingly, the cluster-target correlation now does not seem to be affected by learning while the classification score still increases. But now the variance during pattern presentation is reduced after learning.

4 Discussion

In this study it has been attempted to test hypotheses derived from observations of cortical variability dynamics in the context of reservoir computing. Although it may be counted as a partial success, it can only be the first small step in the quest for the so euphemistically title *Plastic State Machine*.

The first section has confirmed the observation that random recurrent neural networks as computing reservoirs are notoriously hard to control. Allthough an increase in performance over using no reservoir could be achieved with all model-combinations tested, most setups proved to be highly sensitive to the parameter settings in non-obvious ways. Surprisingly, the AS-model, which was never designed as a computing reservoir but simply as a toy network for qualitatively immitating cortical variability, performed best.

The introduction of plasticity in the input has shown that it is possible to improve the performance of this type of computing reservoir through unsupervised learning. It also seems to have confirmed, that selective cluster activation can help classification and that it can be achieved in a self-organising manner. Allthough it may be argued that introducing clusters in the reservoir artificially is biologically justifiable, the aim must be in a next step to somehow make the system learn the clustered connectivity from the data, since there is no straightforward way of choosing the number of clusters and this may strongly affect the systems' performance. Having plasticity inside the reservoir is also essential for the hypotheses mentioned in the introduction that the activities of the clusters in the spontaneous state somehow incode priors of the input stimuli. This is however beyond the scope of this project.

The conclusion of the project is that this may be a promising field for further studies.

References

[1] A. Arieli, A. Sterkin, A. Grinvald, and A. Aertsen. Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses. *Science*, 273(5283):1868, 1996.

[2] Mark M Churchland, Byron M Yu, John P Cunningham, Leo P Sugrue, Marlene R Cohen, Greg S Corrado, William T Newsome, Andrew M Clark, Paymon Hosseini, Benjamin B Scott, David C Bradley, Matthew a Smith, Adam Kohn, J Anthony Movshon, Katherine M Armstrong, Tirin Moore, Steve W Chang, Lawrence H Snyder, Stephen G Lisberger, Nicholas J Priebe, Ian M Finn, David Ferster, Stephen I Ryu, Gopal Santhanam, Maneesh Sahani, and Krishna V Shenoy. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, 13(3):369–78, March 2010.

[3] Anne K Churchland, R Kiani, R Chaudhuri, Xiao-Jing Wang, Alexandre Pouget, and M N Shadlen. Variance as a signature of neural computations during decision making. *Neuron*, 69(4):818–31, February 2011.

[4] Pietro Berkes, Gergo Orbán, Máté Lengyel, and József Fiser. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science (New York, N.Y.)*, 331(6013):83–7, January 2011.

[5] Ashok Litwin-Kumar and Brent Doiron. Slow dynamics and high variability in balanced cortical networks with clustered connections. *Nature neuroscience*, 15(11):1498–1505, September 2012.

[6] W Maass, T Natschläger, and H Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

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

[8] LF Abbott, K Rajan, and H Sompolinsky. Interactions between intrinsic and stimulus-evoked activity in recurrent neural networks. *. . . : an exploration of neuronal . . .* , pages 1–16, 2011.

[9] KD Miller and DJC MacKay. The role of constraints in Hebbian learning. *Neural Computation*, 126:100–126, 1994.